SEGA®
COMPUTER

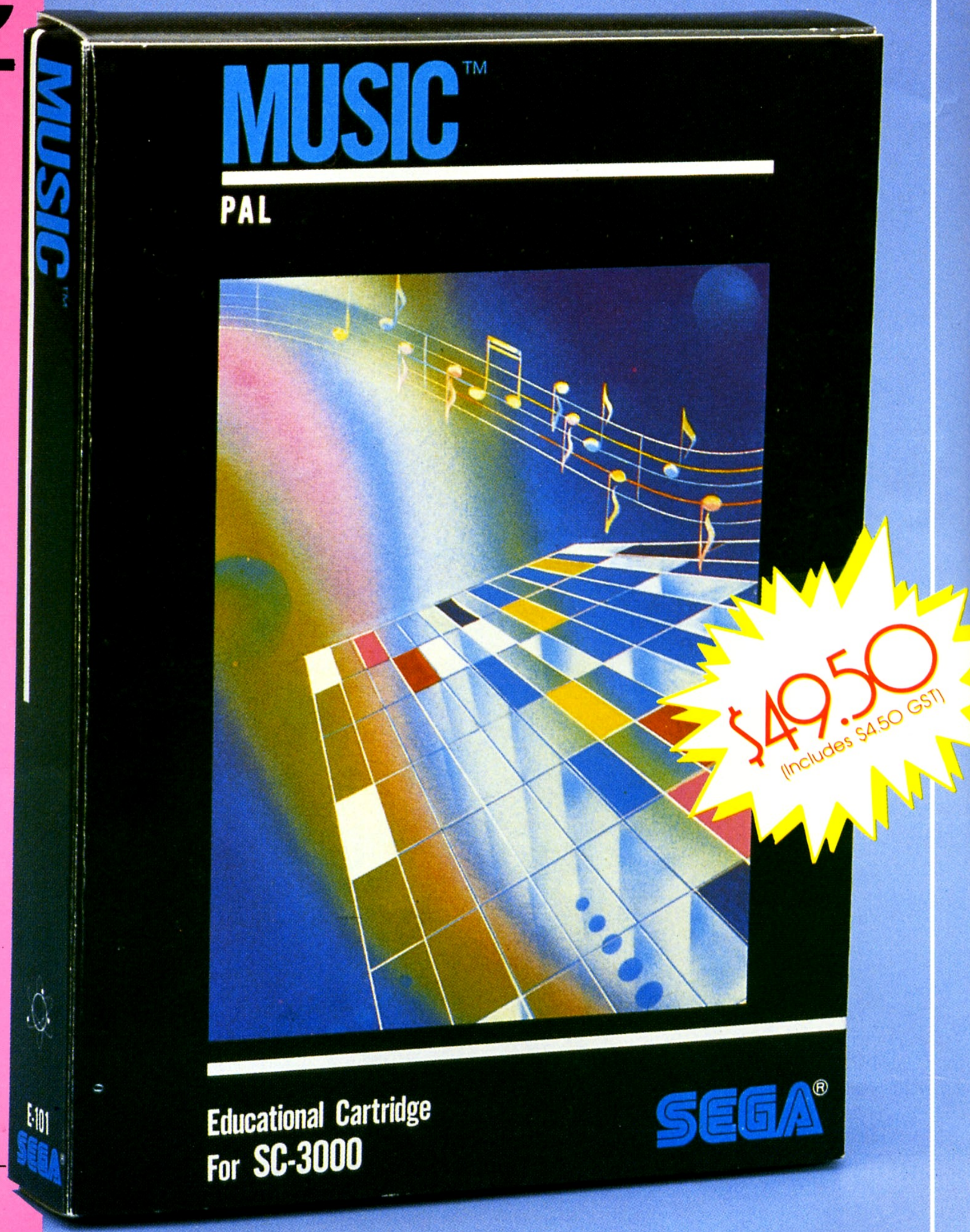
The Official Sega User Club Magazine

SEPT / OCTOBER 1986

In this Issue:
LUNAR TWO
POKER
ALPHA SORTING
NEWS AND REVIEWS
BASIC EXTENSION
PRINTER SHADING
HIGH SCORE CHALLENGE
SEGA PROGRAMMING — Part 2

MUSIC

A fully featured music editor, complete with two keyboard overlays, capable of composition, transcription, changing key signatures, playing as a piano or organ and recording your musical compositions. Fantastic value for all music lovers.



Only Available From

SEGA SOFTWARE SUPPORT

P.O. Box 10-207
Auckland 4
New Zealand

SEGA®
COMPUTER

The official magazine of the Sega User Club, New Zealand.

Registered as a magazine at the CPO Wellington, New Zealand. Published bi-monthly. Magazine published about the middle of July, Sept, Nov, Jan, Mar, May.

Published by:
Sega Software Support
P.O. Box 10-207
Auckland, 4.
NEW ZEALAND

Telephone (09) 603-274

Editor: Glenys Millington

All contributions welcome. Name, address and phone number must be included.

All software programs published by the magazine become the property of Sega Software Support unless by prior arrangement. They are accepted on the basis that they are the original work of the author. The programs must be submitted on tape or disc and a printed listing is desirable. Software is not returned unless accompanied by a stamped self addressed envelope.

For overseas contributions, please enclose a $1 note/coin of the country of origin if software is to be returned.

A plea for software listings - please check your software thoroughly for errors and spelling before sending it to us. Please update us on any errors you know about so that we can publish corrections.

We pay $NZ20.00 for the feature software program published each month and $NZ4.00 for all other full programs published.

Articles or reviews must be legible. Articles or reviews are paid $NZ4.00 per page on a prorata basis.

All contributions are subject to approval by the editor and may be edited to suit the magazine style. All payments are made after publication.

SEGA USER CLUB
MEMBERSHIP YEAR JULY 1986 - JUNE '87

Membership includes a subscription to the Club magazine SEGA COMPUTER as well as qualifying for other club benefits.

NEW ZEALAND
Membership Subscription: $40.70 incl. GST.

Back copies available:
$7.15 each, $35.00 complete 1985/86 year

AUSTRALIA
Membership Subscription:
$NZ50.00 Airmail, $NZ46.00 Surface Mail

ELSEWHERE
Membership Subscriptions available on request.

CONTENTS

5 News and Reviews
6 Lunar Two — Program
13 Sega Programming — Part 2
17 Basic Extension
18 Alpha Sorting — Program
19 Printer Shading
25 Poker — Program
32 Hi-Score Challenge

EDITORIAL

The work just doesn't seem to stop!! Judging by the response, the survey sheet was something all Sega users had been wanting for some time. We hope to have all comments collated by the next issue, and to be able to make comment on some of them. Many remarked on the delays in the magazine and I can assure you that we are trying our best to bring it back into the right time-frame — not an easy task. The competition winners have been notified of their success and our congratulations go to them. The results are printed further on in the mag.

This issue contains Part Two of the serialisation of Brian Brown's book, some more excellent programs, and more interesting information on machine code.

A feather in our cap!! Published recently in the "Bytes and PC's column of Segacious (the Australian user's magazine) was this comment on YOUR magazine:

"Sega Computer, the glossy New Zealand magazine, is now available in Australia! This puts all other Brand-dedicated magazines in the shade! The idiosyncratic writing and programming of Michael Howard, the high-quality readers programs, book reviews and Machine Code lessons make this publication well worth while. Here at Segacious we dream about, one day, being able to produce a glossy of this quality.

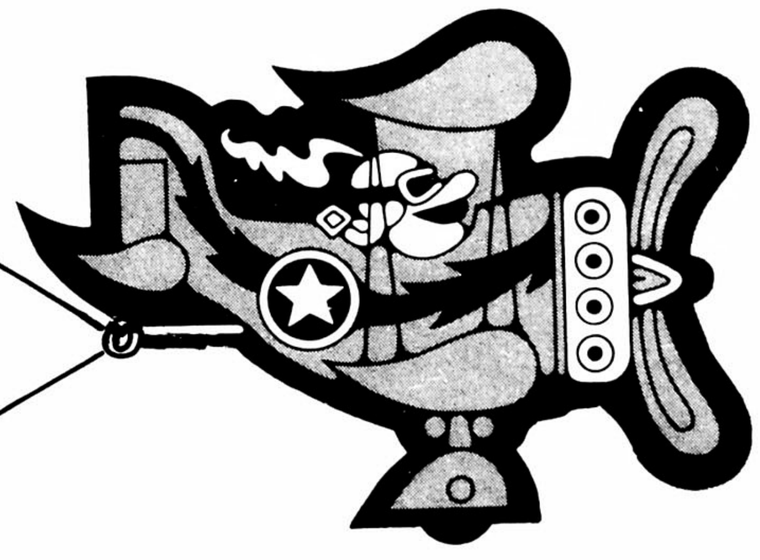
In the meantime, like that second best car rental firm, WE TRY HARDER!!!!

Now after blowing our own trumpets — on with the mag!!!

Glenys Millington
Editor

Glenys Millington

LETTERS ... to THE EDITOR.



DEAR EDITOR
I have the Graph and Chart cassette — how can I get the chart on the printer/plotter? The Graph/Chart program is not much use without that facility.
P.G. Saunders, Auckland.

EDITOR'S REPLY
No idea. Does any one out there know?

DEAR EDITOR
Is it possible to interface an 80 column printer into the Sega and if so what printer will do the job? I appreciate that the Sega is a 40 column computer, but it must be agreed the readout from the Sega printer/plotter leaves a lot to be desired, if using it for letter writing. However it's possible to make 40 columns look more attractive on wider paper.
W.P. Anderson, Christchurch.

EDITOR'S REPLY
Yes, any printer can be interfaced to the Sega. The parallel port (Centronics) is easiest so buy a parallel printer and ask that it has 'Epson compatible" escape codes. Escape codes are the way printers accept information and the 'Epson' type has become the industry standard. The printer cable should be made up at the Sega end according to the information on page 199 of the Sega Control Station Manual. Expect to pay around $600 for the printer and around $50 for the cable. Serial printers are more tricky. Pp. 200-201 contains the information needed and, whoever you buy the printer off, demand that they produce a cable that will work with the Sega.

DEAR EDITOR
I have the game "One Day Cricket". It is a very enjoyable game with great graphics but I am sorry to say it has a number of errors. It is supposed to have fifty, six-ball overs (300 balls), but it only has forty-five, five-ball overs (225 balls), and therefore 75 balls short of what it should be. The other problem is that if you last out your full forty-five overs without losing all your wickets, you cannot go on to the second innings, you have to start all over again. If you could manage to correct these problems, could you please print them in your fab magazine? Also, when will "Exerion" be in stock again?
James Dawson, Howick

EDITOR'S REPLY
Unfortunately the author of One Day Cricket is overseas at present — we will respond on his return, which will not be until Christmastime.
Exerion will never be in stock again as far as we know.

DEAR EDITOR
I enjoyed reading Selwyn Easton's Word Processing reviews and immediately recognised his Sega Word Processor reformatting problem. This caused me similar problems and is a real pain. The problem appears to be in the manual formatting which allows characters to be placed in the righthand most column, when the text is later scrolled with format on, the lines are reformatted. I think manual formatting leaving the righthand column empty cures the problem.
Another problem with this program arises when the "R.H. Justify" is selected from the printer menu. This causes lines of text over half length and longer centred text to be spaced across to the RH column.
This is not generally what was required, but it can be prevented by placing a stop in the second to last column of text on these rows, (this again to prevent reformatting) this holds the text in position and the stops can be snopaked out later if necessary.
Keith Maynard does not mention in his letter if his SP400 redraws the test pattern after returning to the origin. If this is the case, his problem could be a faulty power supply lead, the motion of the plotter could produce intermittant contact. He could check this be wriggling the lead or exchange power supplies with his SC3000, Hope this helps.
P.S. Could Michael Howard please explain -after CBh & — after EDh in his Z80 opcode list.
D. Coursey, Christchurch.

EDITOR'S REPLY
Michael Hadrup on Auckland 534-3379 may be able to help you with your machine code query.

DEAR EDITOR
It is great to see that the magazine offers a questions and answers slot in the form of these letters to you and the subscribing members. I have often read helpful hints for would-be programmers and those experiencing problems with commercial software packages. Some of the solutions offered have themselves taken me considerable time to solve prior to reading them in the magazine and it pleases to see others sharing their knowledge.
In the last magazine (July/Aug 1986) you published my answer to R.J. Thorpe's questions regarding HuCal sorting functions. It was my pleasure to have the opportunity to pass on my explanations to Mr Thorpe. However I'm, not sure that he, and others, will be able to understand the explanation as the all important examples referred to and enclosed with my letter were not printed. Upon reading my reply, as published, I was able to understand the initial explanation but could not follow the explanations of the examples I referred to (and remember . . . I wrote it).

I realise that complete text, as received by you, cannot always be printed in the magazine but feel it is not fair to those who ASK and ANSWER the questions if important related text is edited.
S.J. Easton, Pukekohe.

EDITOR'S REPLY
To Selwyn and others — our apologies — an unfortunate oversight. The examples are printed further on the mag.

DEAR EDITOR
We have owned our Sega computer for about two years. Up until now we have used tapes and cartridges, but we find this frustrating because the tapes we found didn't seem to work. Recently we went into 'Farmers' thinking to buy a disk drive Control Station. We were discouraged from buying it because we were told that Sega accessories such as the printer/plotter and disks would not be available in New Zealand. We are really unwilling to buy a new computer. Could you please advise us.
Jonathan Fletcher, Auckland.

EDITOR'S REPLY
Printers/Plotters and Disks are available right now from Sega Software Support. The disks used by the disk drive unit are the same as used by the Amstrad computer!! There are plenty around.

DEAR EDITOR
I have read with great interest the reviews of the four word processor programs for the Sega computer written by Selwyn Easton and published in the July/August 1986 magazine. I am the happy owner of Basword and I would like to let you know my feelings. This program only costs $28-50 and I feel we are getting real value for money. It is an efficient word processor for home use, simple to use and has all the functions that I would need for typing letters, publicity leaflets, labels and reports etc. I also own a copy of Grandstands 32K Word Processor and have been able to use Segaword for a short period.
I feel that Selwyn Easton gave a good factual report when he reviewed the four word processors but accentuated the disadvantages of Basword too harshly. He gave the impression that Segaword 3 was greatly superior. This I cannot agree with. We Sega users are fortunate to have two good word processors available and the users will naturally have a preference for either one depending on their requirements. The price difference between the two word processors (Basword $28-50 and Segaword $70-90) will also sway some customers on which one they will buy. Taking the price into consideration Basword is my preference.
Segaword is faster in some functions because it has more machine code content but Basword makes up for this because of the extra functions it has built into its program eg. underlining, merging, tabulating, printing multiple copies etc.
I would like to take this opportunity to congratulate you on a very good and interesting magazine. Keep up the good work, including the reviews.
N.A.A. Raynel, Takapuna.

DEAR EDITOR
I'm having trouble with one of the programs from the club magazine.
In the Jan-April issue was a program called 3D Noughts and Crosses. I typed it in, checked several times, and still it doesn't run further than the end of turn 2.
As it begins to draw out the third array showing the places held it always throws up an error message — even though it did the job perfectly the first two times.
At the end of the first line of the array — in other words after having been once through the command 'For J = 1 TO 4' it stops and displays 'Next without FOR Error in 2470'. There appears to be nothing wrong with the sub-routine 2300-2490 — after all it does it correctly twice. There must be some small error in one of the sub-routines to throw this out, but I can't find it. Is anyone else having this problem? Can anyone give me a solution? It is a good game if only I could get it going. Incidentally, I have printed the program out and compared it line by line with the printout in the magazine!
Hoping someone can help.
Alan A. Morgan, Hastings.

EDITOR'S REPLY
We don't have 3D Noughts and Crosses ourselves, on tape. Can anyone out there help.

DEAR EDITOR
A friend has a Brother HR5 printer which I have been trying to operate. The best (and only) result I can get is to print out all the characters with the odd bit of parity. If I use the comset command I then cannot get any response from the printer and the only way to get back to the original is to totally shut down the computer and start up again. Changing the Baud rate on the printer or computer has no noticeable effect.
Another question for you. After playing a game in which the character patterns have altered, it is possible with the Basic IIIB to reset the characters to normal by using the reset button. When using the disk drive this no longer resets the characters — is there some way of doing this, save turning off the machine and resetting again.
K. Nightingale, Whakatane.

EDITOR'S REPLY
Is there someone who has a Brother HR5 who can help with this query? I have not had any experience with this particular printer. Also can anyone help with the query re resetting characters on the disk drive?

DEAR EDITOR
The Sega Overview in the July magazine is most informative, as are the machine code articles. More please.
Regarding B Turnell's query, I think he want's a random shuffle routine for his input letters. (Program printed further on in mag — Ed.)
A couple of queries: I have frequent trouble saving and reloading programs on my dataset (a Euromatic), I've tried all volume settings etc. Any further measures I can take or mods I can make to the dataset?
Is Sega printer output in ASCII code (useable with most printers) or for the Sega printer use only? If the latter, is there any way to output in ASCII code? Possibly someone could do a brief article for the magazine explaining what types of printer can be used with the Sega and what adaptors (if any) need to be used.
Chris Rodliffe, Auckland.

EDITOR'S REPLY
(1) We haven't had a lot of experience on the Euromatic — can anyone make some suggestions?
(2) Refer our answer to W.P. Anderson of Christchurch.

MEMBER-ONLY XMAS SPECIALS!!!!

Christmas is nearly upon us again and we have some great specials lined up for those of you looking for presents for the young (and not so young).

NB: THESE SPECIAL PRICES ONLY APPLY UNTIL DECEMBER 15. GST is included in all prices quoted below, postage and handling extra.

BOOKS

SEGA BEGINNERS GUIDE:	Special Price $11.00
MORE THAN 50 PROGRAMS:	Special Price $11.00
BASIC GAMES PROGRAMMING:	Special Price $11.00

MAGAZINES

BACK ISSUES: Each issue now!	Special Price $3.00
'84-'85 (4 magazines)	Special Price $10.00
'85-'86 (Full Year)	Special Price $12.00

CARTRIDGES

MUSIC: *Always a popular one!*	Special Price $25.00
BASIC IIIB:	Special Price $55.00

Upgrade at this 'never to be repeated' price.

3 GAMES CARTRIDGES OF YOUR CHOICE:
Special Price $55.00

NB: *This offer only applies to those cartridges listed in the catalogue*

1 GAMES CARTRIDGE OF YOUR CHOICE:
Special Price $22.00

CASSETTES

3 EDUCATIONAL CASSETTES OF YOUR CHOICE:
Special Price $15.00

NB. *This offer only applies to the education cassettes listed in the catalogue*

3 BUSINESS & UTILITIES CASSETTES OF YOUR CHOICE:
Special Price $15.00

NB. *Catalogue Numbers 603-609 only*

PERIPHERALS & ACCESSORIES

SP400 PRINTER PLOTTER:	Special Price $225.00
NB. *9V Adaptor included*	
SG1000 CARTRIDGE PLAYER:	Special Price $40.00
NB. *9V Adaptor required*	
JOYSTICK:	Special Price $16.00

DISK SOFTWARE

HUCAL, LOGO, DATABASE: Special Price $55.00

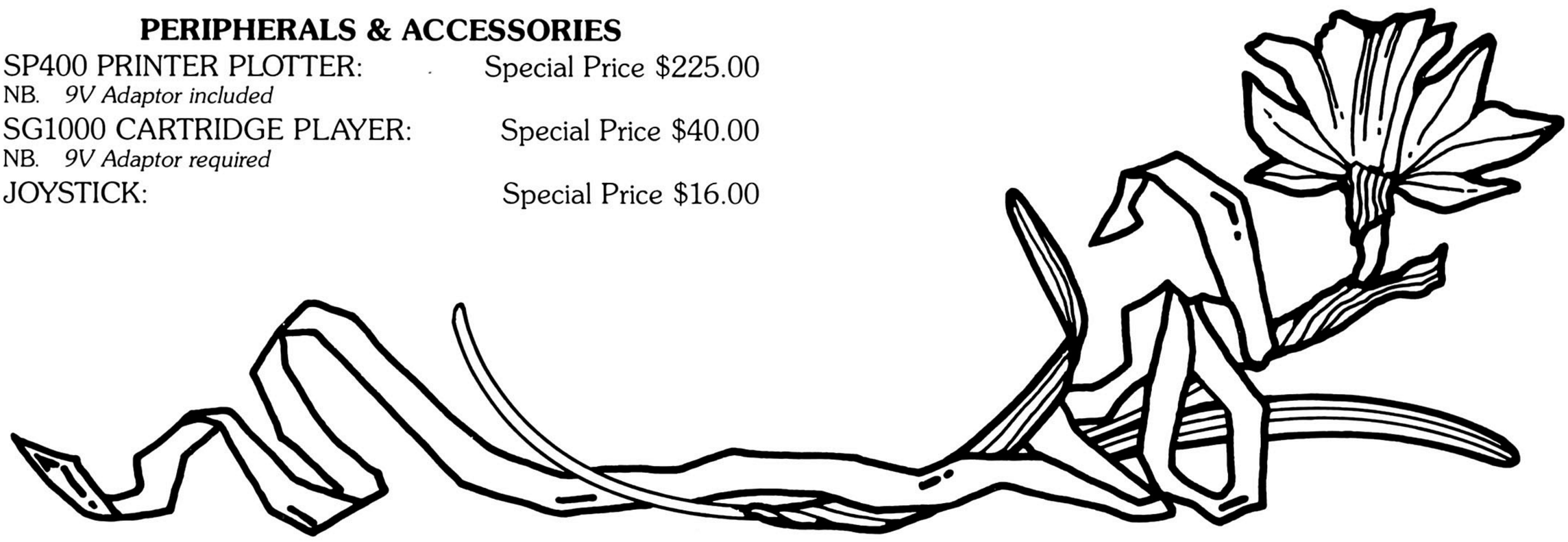
NEW GAMES AVAILABLE:

Now you add to your collection — Arcade packs 2 & 3 are now available for just $30.25 each or buy all three for just $76.00 (incl GST). Three games on each cassette.

Arcade Pack One — Shoot Out, Astro & Gloopa.
Arcade Pack Two — Trojans, Splat & Afos.
Arcade Pack Three — 1986, Blackjack & Bomber.

Great Buying for Christmas!

Another Christmas release — Sir Roderick's Quest. Animated Adventure for the Sega with 64 different screens — test your wits on this one. 32k Only. Price — $30.25



NEWS & REVIEWS

Recently we received word from our counterparts in Australia that Sega in Japan had released the 'My-Card' — an updated version of the cartridge with some excellent games available.

Enquiries in Japan revealed that this was indeed the fact. Just purchase the cards and one card catcher and you're in business. Always on the look-out for new software we also enquired into prices and that is where the excitement stopped. Taking into account the value of the NZ$ against the Japanese yen, 43% Customs Duty and then add on GST and the equivalent of one cartridge would retail in New Zealand for somewhere in the vicinity of $80.00 — $90.00. We were hoping to release the 'My-Cards' prior to Christmas, however we are still negotiating with Japan for a fairer price.

Also released in Japan around the same time as the 'My-Cards' was the Sega Mark III — an update on the SG1000 Cartridge player which uses a card similar to the 'My-Card' instead of the cartridge. Again the yen etc. has proved to be against us all the way. The anticipated retail price for the Mark III would be in the vicinity of $200 — $250.00.

However we live in the hope that negotiations will prove to be successful and that we will still be able to import these products to extend the range of games software currently available.

The big news in Australia is that a version of CP/M for the Sega is now available which will enable users to access programs on Amstrad CP/M Version 2.2 disks. We will update you on this development as soon as we have some further information.

The 'Sega Is Supreme' Competition was a great success with many hundreds of Survey Forms being returned. Four very excited people received telephone calls on the night of October 15 advising them of their luck in the draw.

They were:

1st Prize:
SF7000 Control Station
Karl Woodhead, Hawera

2nd Prize:
Sega Printer Plotter
Mr B.A. Turnell, Waikeria

3rd Prize:
Sega Speech Processor
Paul Englert, Karori, Wellington

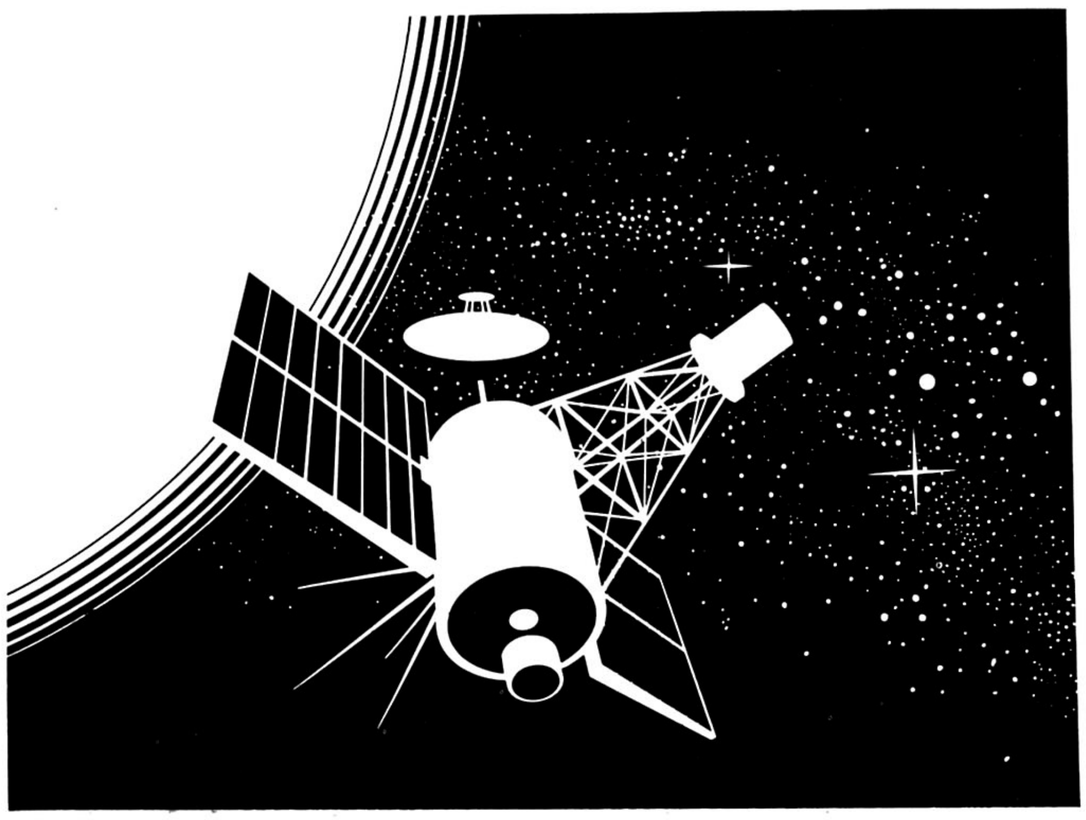
4th Prize:
$100 worth of Software
Raymond Ford, Waiheke Island

Our congratulations to all four winners who will have received their prizes by now, and our thanks to those who returned their survey sheets. We hope to have collated all the comments before the next issue and be able to cover several aspects which were raised.

Glenys Millington

XMAS HOLIDAYS

We will be closed from December 16, 1986, until January 19, 1987. Any orders received during this time will be processed when we re-open. Our compliments of the season to all Sega User Club Members.



100 REM
LUNAR-TWO
...by JOHN DOWMAN

```
100 REM L U N A R - T W O
110 REM By  JOHN DOWMAN
120 REM MISSION 1 was not written by
130 REM JOHN DOWMAN
140 REM ----------------------------
150 REM
160 SCREEN 1,1:COLOR10,1:GOSUB 1870:VD=0:GOTO990
170 REMLOOP
180 IFFU>0THENIFSTRIG(1)>0THENV=V-BN:BLINE(202,59+FU)-(205,60+FU),,B:FU=FU-2:SPR
ITE1,(X,Y-16),4,9:SOUND4,2,7:TP=TP+2:LINE(229,129+TP)-(232,130+TP),8,B:IFTP>50TH
EN710
190 H=(Y-16)*2-4
200 CURSOR218,94:COLOR15:BLINE(224,94)-(251,86),,BF:PRINTINT(H)
210 CURSOR218,70:COLOR15:BLINE(218,70)-(254,62),,BF:PRINTV*2
220 Y=Y-V
230 X=X+(STICK(1)=7)-(STICK(1)=3)
240 IFY>191THENY=191
250 IFSTRIG(1)<1ORFU=0THENSPRITE1,(0,0),0,0
260 IFSTRIG(1)<1ORFU=0THENSOUND0
270 IFX>175THENX=5
280 IFX<5THENX=175
290 SPRITE0,(X,Y),0,10
300 IFINT(Y-16)<=1THEN360
310 IFX<108ORX>120THENIFVPEEK(FNA(A))>0THENGOTO720
320 X=X+(STICK(1)=7)-(STICK(1)=3):V=V+.15
330 TP=TP-.3:BLINE(229,TP+132)-(232,132+TP):IFTP<0THENTP=0
340 GOTO180
350 REMRATING,BONUS
360 SOUND0:IFV>1THEN710
370 CURSOR45,88:COLOR7:PRINT"MISSION ";M;" COMPLETED"
380 CURSOR60,80:PRINT"LANDING VELOCITY":CURSOR60,72:PRINTV*2;" FEET/SEC"
390 FORB=FU+1TO1STEP-1
400 BLINE(202,59+B)-(205,59+B)
410 SOUND0
420 SC=SC+2:BLINE(193,11)-(230,2),,BF:CURSOR188,11:PRINTSC
430 SOUND1,INT(RND(1)*600)+500,10
440 NEXT
450 SOUND0
460 IFV<=.25THENSC=SC+100:BEEP:BEEP:BEEP:BEEP:CURSOR50,140:PRINT"PERFECT LANDING
!":CURSOR15,132:PRINT"YOUR LICENSE WILL BE RENEWED":PRINTSPC(10);"...LATER":GOTO
500
470 IFV<=.5THENSC=SC+70:BEEP:BEEP:BEEP:CURSOR50,140:PRINTA$(INT(RND(1)*3+1));" L
ANDING!":CURSOR45,132:PRINT"BUT STILL NOT PERFECT":CURSOR45,124:PRINT"TAKE IT EA
SY ON THOSE":CURSOR53,116:PRINT"LUNAR LANDING PADS":GOTO500
480 IFV<=.7THENSC=SC+40:BEEP:BEEP:CURSOR50,140:PRINTB$(INT(RND(1)*3+1));" LANDIN
G!":CURSOR25,132:PRINT"BUT A LONG WAY FROM PERFECT":CURSOR30,124:PRINT"YOU BLIND
ED THE MOON MAN":CURSOR40,116:PRINT"WITH YOUR DUST CLOUD":GOTO500
490 IFV<=1THENSC=SC+20:BEEP:CURSOR40,140:PRINT"WHAT A ";C$(INT(RND(1)*2+1));" LA
NDING":CURSOR30,132:PRINT"YOU NEARLY MADE ANOTHER":CURSOR36,124:PRINT"CRATER ON
THE MOON"
500 BLINE(193,11)-(230,2),,BF:CURSOR188,11:PRINTSC
510 BN=BN-.04:IFBN<.25THENBN=.25
520 M=M+1:SFU=SFU-6:FU=SFU:IFSFU<72THENFU=72
```

```
530 LINE(202,60)-(205,59+SFU),10,BF
540 BLINE(229,129)-(232,180),,BF
550 FORD=1TO999:NEXT:BLINE(10,64)-(190,180),,BF
560 CURSOR45,90:PRINT"PREPARE FOR IGNITION"
570 SPRITE2,(X,Y),28,10:SPRITE1,(X,Y),24,10
580 CD=10:CURSOR55,82:PRINT"COUNT DOWN:";CD:TIME$="00:00:00"
590 FORCT=1TO100:IFTIME$="00:00:01"THEN610
600 NEXT
610 CD=CD-1:TIME$="00:00:00"
620 BLINE(121,82)-(140,74),,BF:CURSOR121,82:PRINTCD:BEEP
630 IFCD>0THEN590
640 BLINE (10,64)-(180,180),,BF
650 SPRITE0,(X,Y-16),4,9:SOUND4,2,13
660 Y=191:FORD=1TO5STEP.07
670 Y=Y-D:VPOKE&H3B00,Y:VPOKE&H3B04,Y-16:NEXT:SOUND0
680 SPRITE0,(0,0),,0:SPRITE2,(0,0),,0
690 COLOR7:CURSOR90,180:PRINT"MISSION";M:FORD=1TO500:NEXT:BLINE(10,64)-(180,180)
,,BF:GOTO 2050
700 X=INT(RND(1)*170)+10:Y=191:TP=0:V=2:GOTO170
710 REMEXPLOSION
720 SPRITE1,(X,Y),8,9
730 FORS=15TO11STEP-3:SOUND4,2,S:FORDY=1TO15:NEXT:NEXT
740 SPRITE1,(0,0),0,0
750 SPRITE0,(X,Y),12,9
760 FORS=10TO7STEP-3:SOUND4,2,S:FORDY=1TO15:NEXT:NEXT
770 SPRITE0,(X,Y),16,9
780 FORS=7TO4STEP-3:SOUND4,2,S:FORDY=1TO15:NEXT:NEXT
790 SPRITE0,(X,Y),20,9
800 FORS=4TO0STEP-3:SOUND4,2,S:FORDY=1TO15:NEXT:NEXT
810 SPRITE0,(0,0),0,0
820 SOUND0
830 REMGAME OVER
840 CURSOR65,100
850 FORX=1TO16:SOUND1,INT(RND(1)*600+500),10:PRINTMID$("G A M E  O V E R",X,1);:
SOUND0:NEXT
860 IFSC>HSTHENHS$=N$:HS=SC:CURSOR50,92:PRINT"CONGRADULATIONS A NEW":CURSOR80,84
:PRINT"HI-SCORE!!":BLINE(195,35)-(254,25),,BF:CURSOR188,34:COLOR7:PRINTHS:COLOR1
5:CURSOR231,34:PRINTHS$:FORD=1TO800:NEXT:SPRITE1,(0,0),8,0
870 BLINE(20,120)-(190,65),,BF
880 BN=.56:M=1:X=INT(RND(1)*170)+10:Y=191:V=2:SFU=120:FU=120:TP=0:SC=0
890 LINE(202,60)-(205,179),10,BF
900 BLINE(229,129)-(232,180),,BF
910 CURSOR55,170:COLOR13:PRINT"Expanded by":CURSOR83,162:PRINT" J.F Dowman":CURS
OR87,154:PRINT" 1986"
920 COLOR2:CURSOR65,180:PRINTCHR$(17);"LUNAR #2":PRINTCHR$(16)
930 SPRITE0,(110,17),0,10
940 FORSN=240TO500STEP4:SOUND1,SN,15:SOUND1,-SN+900,15
950 IFSTRIG(1)>0ORINKEY$<>""THEN970
960 NEXT:GOTO940
970 SN=0:SOUND0:BLINE(194,12)-(249,3),,BF:BLINE(30,180)-(190,140),,BF:GOTO1770
980 GOTO1770
990 SCREEN2,2:MAG1
1000 A$(1)="BRILLIANT"
1010 A$(2)="GREAT"
1020 A$(3)="EXELLENT"
1030 B$(1)="GOOD"
1040 B$(2)="NICE"
1050 B$(3)="NOT BAD"
1060 C$(1)="ROUGH"
1070 C$(2)="BUMPY"
1080 POSITION(0,191),0,1
1090 BN=.56:M=1:X=INT(RND(1)*170)+10:Y=191:V=2:SFU=120:FU=120:TP=0:SC=0:HS=0
1100 COLOR10,1,(0,0)-(0,0),1:CLS
1110 LINE(191,0)-(255,191),4,B:BLINE(255,64)-(255,71):PATTERNS#255,"808080808080
8080":SPRITE31,(247,80),255,4
1120 REMGAUGE
1130 LINE(202,60)-(205,179),10,BF
1140 LINE(200,55)-(217,185),10,B
1150 FORF=466TO469:READA:VPOKEF,A:NEXT
1160 FORF=2260TO2263:READA:VPOKEF,A:NEXT
```

```
1170 FORF=4554TO4558:READA:VPOKEF,A:NEXT:FORF=4562TO4566:READA:VPOKEF,A:NEXT:FOR
F=4570TO4574:READA:VPOKEF,A:NEXT
1180 REM TEMP GAUGE
1190 LINE(227,185)-(245,125),8,B
1200 COLOR8
1210 FORT=491TO494:READA:VPOKET,A:NEXT:FORT=499TO502:READA:VPOKET,A:NEXT
1220 FORT=2027TO2030:READA:VPOKET,A:NEXT
1230 FORT=2275TO2279:READA:VPOKET,A:NEXT:FORT=2283TO2287:READA:VPOKET,A:NEXT:FOR
T=2291TO2295:READA:VPOKET,A:NEXT:FORC=10467TO10487:VPOKEC,129:NEXT
1240 REM ALT & VEL
1250 FORB=3040TO3055:READA:VPOKEB,A:NEXT
1260 FORB=3808TO3823:READA:VPOKEB,A:NEXT
1270 REM CREATE SURFACE
1280 FORP=0TO180STEP12
1290 READD1,D2
1300 LINE(P,D1)-(P+12,D2),12
1310 NEXT
1320 PRESET(192,21)
1330 REM SCORE
1340 LINE(192,45)-(255,45),4
1350 LINE(192,23)-(255,23),4
1360 CURSOR201,43:COLOR12:PRINT"HI-SCORE"
1370 CURSOR210,21:COLOR12:PRINT"SCORE"
1380 DEF FNA(A)=INT((210/Y)/8)*256+INT((X+8)/8)*8+(210-Y)MOD8
1390 REM CREATE SPRITES
1400 REM LANDER
1410 PATTERNS#0,"0000000001020404"
1420 PATTERNS#1,"0702030F0B171138"
1430 PATTERNS#2,"00000000C0A09090"
1440 PATTERNS#3,"F0A0E0F8E8F4C40E"
1450 PATTERNS#24,"0000000001020404"
1460 PATTERNS#25,"0702030103070 1"
1470 PATTERNS#26,"00000000C0A09090"
1480 PATTERNS#27,"F0A0E0C0E0F0C0"
1490 PATTERNS#29,"0000030F0B171138"
1500 PATTERNS#31,"0000E0F8E8F4C40E"
1510 REM THRUST
1520 PATTERNS#4,"0103050B090D1305"
1530 PATTERNS#5,"0102"
1540 PATTERNS#6,"C0E0D0C8C8D0C0E8"
1550 PATTERNS#7,"80A08080"
1560 REM EXPLOSION
1570 PATTERNS#8,"0000000005000404"
1580 PATTERNS#9,"09031503152761 0"
1590 PATTERNS#10,"00000000C0A030AC"
1600 PATTERNS#11,"4030A0E898B8C40E"
1610 REM
1620 PATTERNS#12,"0000100108000414"
1630 PATTERNS#13,"004A002D035A6521"
1640 PATTERNS#14,"00000000D08000AC"
1650 PATTERNS#15,"00A04890B050C80D"
1660 REM
1670 PATTERNS#16,"0020020040080400"
1680 PATTERNS#17,"2104002A812B4042"
1690 PATTERNS#18,"0040808800A00022"
1700 PATTERNS#19,"044028A052C00420"
1710 REM
1720 PATTERNS#20,"0020000048000004"
1730 PATTERNS#21,"0000108104000 0A1"
1740 PATTERNS#22,"0100000800002000"
1750 PATTERNS#23,"8000880440080022"
1760 REM BEGIN GAME
1770 SCREEN1,1:COLOR10,1:CLS:CURSOR8,10:PRINT"ENTER YOUR INITIALS"
1780 CURSOR10,12:INPUTN$:N$=LEFT$(N$,3)
1790 SCREEN2,2:CURSOR231,11:COLOR15:PRINTN$
1800 GOTO170
1810 DATA 178,149,165,178,200,148,212,200,246,134,228,196,195,189,161,185,177,61
,0,0,128,128,224
1820 DATA 36,42,42,36,68,164,164,68,16,40,40,16
1830 DATA 30,8,8,12,12,242,133,229,197,245,156,82,92,88,88
```

```
1840 DATA 114,138,138,138,251,203,203,203,31,4,4,4,4,6,6,230
1850 DATA 203,202,202,139,139,83,83,35,208,16,16,144,24,24,24,223
1860 DATA 21,16,16,33,33,62,62,57,57,36,36,22,22,15,15,3,3,0,0,0,0,0,0,4,4,22,22
,33,33,23,23,21
1870 RESTORE:SCREEN 2,2:COLOR10,1,,1:CLS
1880 CURSOR80,60:PRINTCHR$(17);"LUNAR #2":PRINT,,,,,,CHR$(16);SPC(15);"Expanded
by John Dowman"
1890 CURSOR30,140:PRINT"Press any key or the fire button":CURSOR100,155:PRINT"to
 start"
1900 FORSN=240TO500STEP4:SOUND1,SN,10:SOUND1,-SN+900,10
1910 IFSTRIG(1)>0ORINKEY$<>""THEN1940
1920 NEXT:GOTO1900
1930 SN=0:SOUND0:BLINE(194,12)-(249,3),,BF:BLINE(80,180)-(190,140),,BF:GOTO1770
1940 REM IF YOU HAVE THE 16k VERSION  THEN YOU CAN DELETE 1840-1920 AND PUT AS L
INE No. 1840 RETURN
1950 REM I THINK THAT YOU HAD BETTER  READ THE INSTRUCTIONS ANY WAY.
1960 SOUND0:CLS:CURSOR50,50:PRINT "Are intructions required{Y/N}?"
1970 IFINKEY$<>"Y"ANDINKEY$<>"N"THEN1970
1980 IF INKEY$="N" THEN RETURN
1990 CLS:SCREEN1,1:CLS:PRINTSPC(10);"Intructions":PRINT"In the first screen, you
 must land on the planet using the Joy-Stick to moveand the fire button for thru
st. Make  sure that the VEL guage is under 1.9  or else you will crash and be fi
nished."
2000 PRINT"In the second screen you are flying  in a mine field.  To destroy th
e minesyou must press the same number as  there are mines, but you only have
a  limited amount of time and you can  only take 5 collisions into the mines"
2010 PRINT:PRINT"In the third screen you are on a red  planet and are fighting o
ff aliens whoare trying to destroy you and you can only take 50 laser hits and t
hen you  DIE.  If their beam is white then you are hit.  The laser comes from sp
ace, too."
2020 PRINT:PRINT"Use the Joy-Stick and the fire button"
2030 IF INKEY$="" THEN 2030
2040 RETURN
2050 SCREEN 2,2:COLOR,8,,8:CLS:COLOR,1,,1:CLS:MAG1
2060 PATTERNC#ASC("@"),"C3DB3C3CDBC300":O=0:S=0
2070 POSITION (0,0),0,0
2080 SPRITE0,(120,150),0,10
2090 SPRITE1,(120,166),4,6
2100 PATTERNS#24,"0408180661992 5DA"
2110 PATTERNS#25,"DA259961061808 04"
2120 PATTERNS#26,"201018608699A45B"
2130 PATTERNS#27,"5BA49986601810 20"
2140 SPRITE2,(40,40),24,7
2150 REM MISSION 2
2160 SOUND4,2,10:FOR G=1 TO 10
2170 BLINE (0,0)-(255,191),,BF
2180 N=INT(RND(1)*9+1)
2190 FOR I=1 TO N
2200 IF I<>1 AND I<>4 THEN 2220
2210 PRINT
2220 COLOR2:CURSORINT(RND(1)*100+50),INT(RND(1)*50+50):PRINTCHR$(16);"@"
2230 PRINT
2240 NEXT I
2250 PRINT
2260 FOR I=1 TO 100
2270 Q=VAL("0"+INKEY$)
2280 IF Q<>0 THEN 2320
2290 NEXT I
2300 SCREEN 1,1:CLS:CURSOR10,10:PRINT "CRASHED INTO A MINE.":O=O+1:FOR L=1 TO 30
0:NEXT L:SCREEN 2,2:IF O>5 THENXX=125:Y=160:GOTO 3540
2310 GOTO 2360
2320 IF Q<>N THEN 2340
2330 SCREEN 1,1:CLS:CURSOR10,10:PRINT"YOU DESTROYED IT!!":FOR L=1 TO 300:NEXT L:
S=S+1:SCREEN 2,2
2340 IF Q<N THEN SCREEN 1,1:CLS:CURSOR10,10:PRINT"NOT STRONG ENOUGH":PRINT:PRINT
"YOU CRASHED INTO A MINE":O=O+1:FOR L=1 TO 300:NEXT L:SCREEN 2,2:IF O>5 THENXX=1
25:Y=160:GOTO 3540
2350 IF Q>N THEN SCREEN 1,1:CLS:CURSOR10,10:PRINT"TOO STRONG":PRINT:PRINT"YOU CR
ASHED INTO A MINE":O=O+1:FOR I=1 TO 300:NEXT I:SCREEN 2,2:IF O>5 THEN XX=125:Y=1
60:GOTO 3540
```

```
2360 NEXT G:SOUNDO
2370 SCREEN 1,1:CLS:CURSORO,10:PRINT"YOU HIT";S;" OUT OF 10"
2380 PRINT:PRINT" BONUS POINTS:-";INT(S*5)
2390 SC=SC+INT(S*5)
2400 PRINT:PRINT"YOUR SCORE IS:-";SC
2410 PRINT:PRINT"GOOD LUCK ";N$
2420 PRINT "                        PRESS ANY KEY..."
2430 IF INKEY$="" THEN 2430
2440 PRINT
2450 PRINT "MISSION 3"
2460 FOR I=1 TO 500
2470 NEXT I
2480 SCREEN 1,1:CLS
2490 SCREEN 2,2:COLOR,8,,8:CLS
2500 COLOR,1,,1
2510 CLS
2520 REM MISSION 3
2530 MAG1:VD=0:X=120:Y=170:SPRITEO,(X,Y),0,10:SPRITE1,(120,10),25,4
2540 LINE (0,0)-(255,191),13,B:LINE (1,1)-(254,190),,B:LINE (2,2)-(254,189),,B:L
INE (3,3)-(253,188),,B
2550 SOUND4,2,15:FOR Y=170 TO 10 STEP -3:SPRITEO,(X,Y),0,10:SPRITE2,(X,Y+16),4,8
:NEXT Y:SPRITEO,(X,10),0,10:SOUNDO:SPRITEO,(0,0),100,0:SPRITE2,(0,0),100,0
2560 FOR I=1 TO 200:NEXT I:SPRITE1,(0,0),25,0
2570 SOUNDO
2580 SPRITE1,(0,0),255,0
2590 MAG1
2600 SPRITEO,(115,160),0,10
2610 SPRITE1,(115,176),4,8
2620 SOUND4,2,10
2630 X=INT(RND(1)*150)+50
2640 X1=INT(RND(1)*150)+50
2650 XX=INT(RND(1)*150)+50
2660 C=INT(RND(1)*13)+2
2670 C1=INT(RND(1)*13)+2
2680 CC=INT(RND(1)*13)+2
2690 FOR Y=0 TO 160 STEP 2
2700 SPRITE2,(X,Y),0,C
2710 SPRITE3,(X,Y+16),4,8
2720 SPRITE4,(X1,Y),0,C1
2730 SPRITE5,(X1,Y+16),4,8
2740 SPRITE6,(XX,Y),0,CC
2750 SPRITE7,(XX,Y+16),4,8
2760 NEXT Y:XX=115
2770 SCREEN 2,2:CLS:SPRITEO,(XX,160),0,10
2780 SPRITE1,(115,176),4,8
2790 FOR I=15 TO 0 STEP -.05:SOUND4,,I:NEXT:SPRITE1,(0,0),4,0
2800 CIRCLE(127,191),100,8,.15,.5,1,BF
2810 CURSOR25,100
2820 FOR I=1 TO 16:SOUND1,INT(RND(1)*600+500),13:PRINTCHR$(17);MID$("P R E S S "
"T""",I,1);:SOUNDO:NEXTI:PRINT CHR$(16):CURSOR40,120:PRINT "TO CONTINUE"
2830 FOR U=1 TO 100:NEXTU
2840 BLINE (25,100)-(255,130),,BF
2850 IF INKEY$<>"T" THEN 2850
2860 PATTERNS#36,"00002090C8A9EFFB"
2870 PATTERNS#37,"F8EFA9C89020"
2880 PATTERNS#38,"000000000070F8EF"
2890 PATTERNS#39,"EFF870"
2900 PATTERNS#40,"080C2E7F2E0E4AFF"
2910 PATTERNS#41,"752F5FB76B67FF55"
2920 PATTERNS#42,"001C08484B4A6AFF"
2930 PATTERNS#43,"F7FEFCFC78D8FF55"
2940 FOR PO=1 TO 50
2950 K=0:LL=INT(RND(-1)*2)+1
2960 IF LL=1 THEN3000
2970 IF LL=2 THEN3100
2980 NEXTPO
2990 GOTO 3750
3000 Y=INT(RND(1)*50)+20:C=INT(RND(1)*13+2):A=INT(RND(1)*10+10)
3010 FOR X=10 TO 235 STEP A
3020 SPRITE1,(X,Y),36,C
3030 Q=INT(RND(8)*11)+1
```

```
3040 ONQGOSUB3380,3460,3420,3460,3460,3380,3400,3460,3380,3460,3470
3050 IF STRIG(1)=1 THEN GOSUB 3200
3060 IFSTICK(1)=3ANDXX<220THENXX=XX+5
3070 IFSTICK(1)=7ANDXX>30THENXX=XX-5
3080 SPRITE0,(XX,160),0,10:NEXT X
3090 GOTO 2980
3100 Y=INT(RND(1)*50)+20:C=INT(RND(1)*13+2)
3110 FOR X=235 TO 10 STEP -INT(RND(1)*10+10)
3120 SPRITE1,(X,Y),40,C
3130 IF STRIG(1)=1 THEN GOSUB 3200
3140 Q=INT(RND(1)*11)+1
3150 ONQGOSUB3380,3460,3420,3460,3460,3380,3400,3460,3380,3460,3470
3160 IFSTICK(1)=3ANDXX<220THENXX=XX+5
3170 IFSTICK(1)=7ANDXX>30THENXX=XX-5
3180 SPRITE0,(XX,160),0,10:NEXT X
3190 GOTO 2980
3200 LINE (XX+8,161)-(XX+8,0),12:SOUND5,RND(1)*3,15:FOR I=1 TO 5:NEXT I:SOUND0:B
LINE (XX+8,161)-(XX+8,0)
3210 IFXX+8>XANDXX+8<X+16THENGOSUB3230
3220 RETURN
3230 REM EXPLOSION
3240 FOR S=15TO11STEP-3:SOUND4,2,S:FORDY=1TO15:NEXT:NEXT
3250 SPRITE1,(0,0),0,0
3260 SPRITE1,(X,Y),12,9
3270 FOR S=10TO7 STEP -3:SOUND4,2,S:FOR DY=1TO15:NEXT:NEXT
3280 SPRITE1,(X,Y),16,9
3290 FOR S=7 TO 4 STEP -3:SOUND4,2,S:FOR DY=1TO15:NEXT:NEXT
3300 SPRITE1,(X,Y),20,9
3310 FOR S=4 TO 0 STEP -3:SOUND4,2,S:FOR DY=1 TO 15:NEXT :NEXT
3320 SPRITE1,(0,0),0,0
3330 SOUND0
3340 IF LL=1 THEN X=235:GOTO 3360
3350 X=10
3360 Z=INT(RND(1)*20+20):SC=SC+Z
3370 RETURN
3380 LINE (X,Y)-(XX-9,160),8:SOUND4,RND(1)*3,15:BLINE (X,Y)-(XX-9,160):SOUND0
3390 RETURN
3400 LINE (X,Y)-(XX+25,160),8:SOUND4,RND(1)*3,15:BLINE (X,Y)-(XX+25,160):SOUND0
3410 SOUND0:RETURN
3420 X1=INT(RND(1)*255)
3430 LINE (0,0)-(X1,160),RND(1)*13+2:LINE (255,0)-(X1,160):SOUND4,RND(1)*3,15:BL
INE (0,0)-(X1,160):BLINE (255,0)-(X1,160):SOUND0
3440 IF X1>XXANDX1<XX+16THENGOSUB3490
3450 RETURN
3460 RETURN
3470 LINE (X+8,Y)-(XX+8,166),15:SOUND4,RND(1)*3,15:BLINE (X+8,Y)-(XX+8,166):SOUN
D0
3480 GOSUB 3490:RETURN
3490 SCREEN 1,1:CLS:VD=VD+1
3500 IF VD>50 THEN 3540
3510 CURSOR5,8:PRINT "You have been hit":PRINT:PRINT"You can only take";50-VD;"
more direct hits!"
3520 FOR I=1 TO 300:NEXT
3530 SCREEN 2,2:RETURN
3540 SCREEN 2,2:SPRITE0,(XX,160),8,8
3550 FOR S=15 TO 11 STEP -3:SOUND4,2,S:FOR DY=1 TO 15:NEXT :NEXT
3560 SPRITE2,(0,0),0,0
3570 SPRITE0,(XX,160),12,9
3580 FOR S=10 TO 7 STEP -3:SOUND4,2,S:FOR DY=1 TO 15:NEXT :NEXT
3590 SPRITE0,(XX,160),16,8
3600 FOR S=7 TO 4 STEP -3:SOUND4,2,S:FOR DY=1 TO 15:NEXT :NEXT
3610 SPRITE0,(XX,160),20,8
3620 FOR S=4 TO 0 STEP -3:SOUND4,2,S:FOR DY=1 TO 15:NEXT :NEXT
3630 SPRITE0,(0,0),0,0
3640 SOUND0
3650 REM GAME OVER
3660 CURSOR65,100
3670 FOR X2=1 TO 16:SOUND1,INT(RND(1)*600+500),10:PRINT MID$("G A M E  O V E R",
X2,1);:SOUND0:NEXT X2:CURSOR75,150:PRINT "SCORE:-";SC
3680 IF SC>HS THEN HS$=N$:HS=SC
```

```
3690 FOR M=1 TO 1000:NEXTM
3700 BLINE (0,0)-(255,191),,BF
3710 CURSOR50,50:PRINT "Do you want to play again ?(Y/N)"
3720 IF INKEY$="" THEN 3720
3730 IF INKEY$="Y"ORINKEY$="y"THEN100
3740 IF INKEY$="N"ORINKEY$="n"THENSCREEN 1,1:CLS:CURSOR10,15:PRINT "See you late
r...":END
3750 SCREEN 1,1:CLS
3760 SOUND0
3770 CURSOR0,10:PRINT "MISSION 3 COMPLETE."
3780 PRINT:PRINT "YOUR SCORE";SC
3790 PRINT:PRINT "CONGRADULATIONS ";N$;" !!
3800 FOR SS=1 TO 5
3810 FOR S=110 TO 1000 STEP 30
3820 SOUND1,S,10:NEXTS:NEXT SS
3830 SOUND0
3840 PRINT :PRINT "Do you want to play again?(Y/N)
3850 IF INKEY$="" THEN3850
3860 IF INKEY$="Y" THEN 160
3870 CLS:CURSOR15,15:PRINT "See you later ";N$:END
```

Cont'd from page 2

Example 1

COLUMN A	COLUMN B	COLUMN C
1.	ANDREW	SMITH
2.	STEVEN	JONES
3.	RICHARD	EDWARDS
4.	BILL	BAXTER
5.	ANDREW	EDWARDS

Example 2

COLUMN A	COLUMN B	COLUMN C
5.	ANDREW	EDWARDS
4.	BILL	BAXTER
3.	RICHARD	EDWARDS
2.	STEVEN	JONES
1.	ANDREW	SMITH

Example 3

COLUMN A	COLUMN B	COLUMN C
5.	ANDREW	EDWARDS
1.	ANDREW	SMITH
4.	BILL	BAXTER
3.	RICHARD	EDWARDS
2.	STEVEN	JONES

Example 4

COLUMN A	COLUMN B	COLUMN C
4.	BILL	BAXTER
5.	ANDREW	EDWARDS
3.	RICHARD	EDWARDS
2.	STEVEN	JONES
1.	ANDREW	SMITH

Example 5

COLUMN A	COLUMN B	COLUMN C
5.	ANDREW	EDWARDS
1.	ANDREW	SMITH
4.	BILL	BAXTER
3.	RICHARD	EDWARDS
2.	STEVEN	JONES

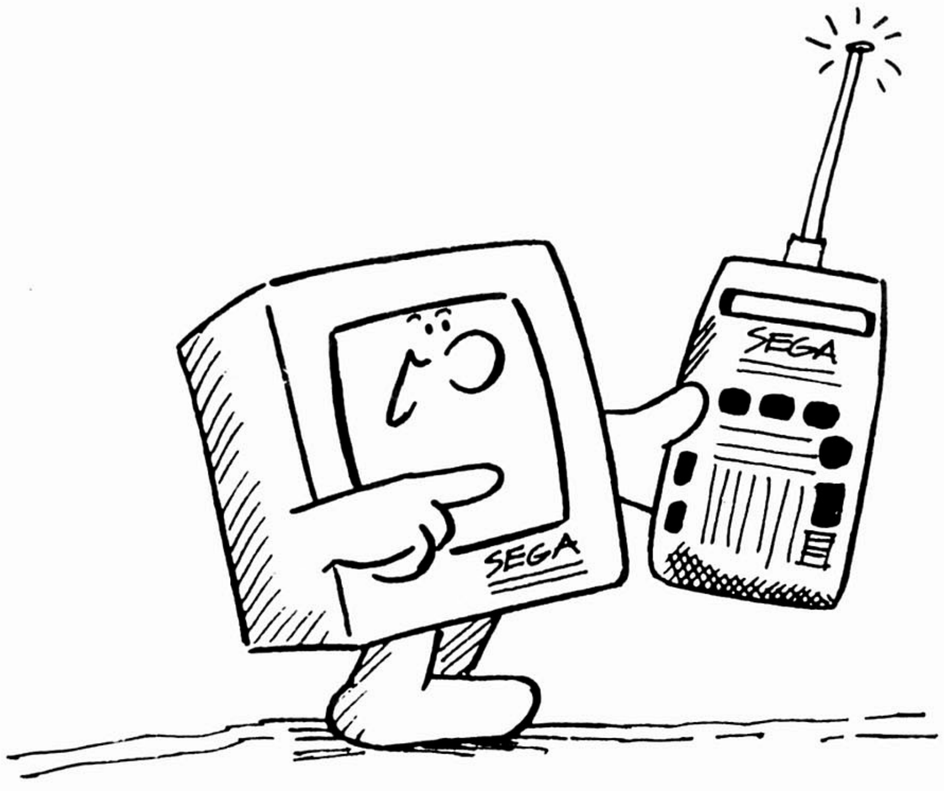
Example 6

COLUMN A	COLUMN B	COLUMN C
4.	BILL	BAXTER
5.	ANDREW	EDWARDS
3.	RICHARD	EDWARDS
2.	STEVEN	JONES
1.	ANDREW	SMITH

Cont'd from page 3

Sega Random-shuffle Routine

```
 10 INPUT I$
 20 LI=LEN(I$)
 30 DIM J$(LI)
 40 FOR K=1 TO LI
 50 J=INT(RND(-1) * LEN(I$)) +1
 60 J$(K)=MID$(I$,J,1)
 70 IL$="": IF J > 1 THEN IL$ = LEFT$(I$,J-1)
 80 IR$="": IF LEN(I$) > J THEN IR$+RIGHT$(I$,LEN(I$)-J)
 90 I$=IL$ + IR$
100 NEXT K
110 M$=""
120 FOR K+1 TO LI
130 M$=M$ = J$(K)
140 NEXT K
150 PRINT M$
```



Part 2
The SEGA Computer
THE VISUAL DISPLAY PROCESSOR

by Brian Brown

The VDP is a Texas Instruments 9929A chip. This has several important features, such as sprites and interrupt capabilities. In the SEGA computer, the VDP is mapped at two port locations, &HBE and &HBF. These ports are the means by which the central processor communicates with the VDP chip and the Video RAM.

THE VISUAL DISPLAY MODES

The VDP has four separate display modes. The four modes are:
(1) Graphics Mode I
(2) Graphics Mode II
(3) Text Mode
(4) Multicolor Mode

Only the two used in the SEGA will be explained here, but a program which allows the user to program the multicolor mode is appended at the end of this chapter.

THE TEXT MODE:

The text mode provides for 40 characters wide by 24 lines of text. Only two colors may be present on the screen at any time. BASIC only allows the use of 38 characters per line, this is done to allow for older television sets which might chop off the first two characters. The two colors referred to as the writing or foreground color, and the background color. These colors are specified by the COLOR command, or may be altered by poking location &H9339 with the appropriate value. The address of the Video RAM (VRAM) used to store the characters is as shown.

```
      ********************************************
3000  *                                          * 3C27
      *                                          *
      *                                          *
3F98  *                                          * 3FBF
      ********************************************
```

THE GRAPHICS MODE II:

The graphics mode allows all 16 colors to be used to be used simultaneously, and the display is arranged as 256 by 192 pixels, where a PIXEL is a single dot on the screen. A separate area in VRAM is used to store the colour attribute of each pattern on the screen. The patterns are stored as follows, (displaying the 1st character in line 1 only)

```
*  ********  *
*    0000    *
*    0001    *
*    0002    *
*    0003    *
*    0004    *
*    0005    *
*    0006    *
*    0007    *
*  ********  *
```

The characters are normally made up out of 8x8 pixel blocks. This shows the makeup of the first character of the first line on the graphics screen. The eight bytes that make the character are arranged as shown, with the address enclosed. The second character will thus use address's 0008 " 0017 etc. The color byte for each character is located at &H2000 ", ie, the color attribute address for the 1st byte is &H2000, for the second byte it is &H2001, for the 1st byte of the 2nd character it is &H2008.

The following program illustrates the colours available in the graphics mode.

```
10 SCREEN 2,2:CLS:B=0
20 FOR X=&HOOOO TO &H17FF
30 B=B+1:IF B=7 THEN RESTORE:B=0
40 READ A:VPOKE X,&HFO
50 VPOKE X+&H2000,A
60 NEXT X
70 GOTO 70
80 DATA &HO1,&H24,&H35,&H6A,&H7B
90 DATA &H8C,&H9D,&HEF
```

ARCHITECTURE OF THE TMMS9929A:

The VDP chip comprises eight (8 bit) write only registers, a read only (8 bit) status register, and an autoincrementing (14 bit) address register. The registers hold the necessary addresses or data for the VDP chip to be able to find the required patterns in the VRAM and determine the location, color, size etc of the sprites or the text. The eight register functions in turn are:

```
Register 0: Register 0 controls the external VDP input, as
            well as mode select. The external VDP input
            allows the image from another VDP to appear in
            the background.  In the case of the SEGA this is
            disabled. MODE SELECT(M3) controls the format of
            the display screen. This is combined with the M2
            and M1 of the register 1 to select the desired
            screen layout. (See Table XXX5)

   Bits       7    6    5    4    3    2    1    0
            ****************************************

            * 0  * 0  * 0  * 0  * 0  * 0  * M3* EV *

            ****************************************

Register 1: Register 1 controls the Video RAM type selection,
            the blanking out of the active display area,
            interrupt enable, M1, M2 and the size and
            magnification factor of any sprites. The SEGA
            computer has the following, VRAM bit = 1 for 4116
            type, Blank bit = 1, Interrupt enabled(50Hz) = 1,
            Screen mode = text, Size and Mag are 0.
            SIZE: This bit determines whether 8 x 8 sprites
            or 16 x 16 sprites are used.
            MAG: This doubles the size of the sprites if a 1,
            else if a 0 the the size is that set by the size
            bit. (Table XXX6 gives the combinations equal to
            the MAG command on the SEGA).

   Bits       7     6     5     4     3     2     1     0
            ************************************************

            * 16k*  BL * IE  * M1   * M2  * 0  *SIZ*  MAG *

            ************************************************

Register 2: Register 2 holds the NAME TABLE address for the
            text or graphic screen, this being &H3000(text)
            or &H0000(graphics).

   Bits       7    6    5    4    3    2    1    0
            ****************************************

            * 0   * 0  * 0  * 0  * 4 bit address    *

            ****************************************
            Actual Address = 4 bit address * &H400
```

```
Register 3: Register 3 holds the COLOUR ATTRIBUTE TABLE
            address for the graphics screen, this equal to
            the &H2000 for the SEGA.

   Bits        7    6    5    4    3    2    1    0
            ****************************************

            *            8 bit address               *

            ****************************************
            Actual Address = 8 bit address * &H40


Register 4: Register 4 holds the PATTERN GENERATOR address
            for the text or graphic screen, being
            &H1800(text) or &H3800(graphics).

   Bits        7    6    5    4    3    2    1    0
            ****************************************

             * 0  * 0  * 0  * 0  * 0  * 3 bit add   *

            ****************************************
            Actual Address = 3 bit address * &H800


Register 5: Register 5 holds the SPRITE ATTRIBUTE address
            (&H800).

   Bits        7    6    5    4    3    2    1    0
            ****************************************

            *  0 *        6 bit address              *

            ****************************************
            Actual Address = 6 bit address * &H80


Register 6: Register 6 holds the SPRITE PATTERN address
            (&H1800).

   Bits.       7    6    5    4    3    2    1    0
            ****************************************

             * 0  * 0  * 0  * 0  * 0  *  3 bit add  *

            ****************************************
            Actual Address = 3 bit address * &H800


Register 7: Register 7 holds the COLOR for the
            writing/background combination.

   Bits        7    6    5    4    3    2    1    0
            ****************************************

            * Writing Color     * Background Color  *

            ****************************************


STATUS
Register:   The status register holds the interrupt flag, the
            fifth sprite flag and number, and the sprite
            collision flag.

   Bits        7    6    5    4    3    2    1    0
            ****************************************

             * F  * 5S * C  *   Fifth Sprite Num    *

            ****************************************
```

HOW TO WRITE/UPDATE A VDP REGISTER:

Two bytes are required to update or write to a register.

Byte 1 is the required data
Byte 2 is the required register

The composition of byte 2 is:

10000 + RSO + RS1 + RS2 (Where RSO-2 are 1 bit each)

	RS0	RS1	RS2
Register 0	0	0	0
Register 1	0	0	1
Register 2	0	1	0
Register 3	0	1	1
Register 4	1	0	0
Register 5	1	0	1
Register 6	1	1	0
Register 7	1	1	1

NOTE: IT IS IMPORTANT THAT THE STATUS REGISTER IS READ AT PORT &HBF BEFORE YOU UPDATE ANY VDP REGISTER.

There is a ROM routine at &H2C54 which provides this facility. Load Register C with the register number (0"7), Register A with the Data byte before calling.

EXAMPLE: Change the color information of the text screen by directly writing to the VDP register 7.

```
 10 SCREEN 1,1:CLS
 20 PRINT "This is actually black writing"
 30 PRINT "on a green background."
 40 FOR X = &HA000 TO &HA00C
 50 READ AA : POKE X,AA : NEXT DE
 60 FOR DE = 1 TO 500 : NEXT DE
 70 CALL &HA000 : PRINT "But is it really?"
 80 GOTO 80
 90 DATA 243,219,191,62,33,211,191
100 DATA 62,135,211,191,251,201
110 REM Disable interrupts, read status register
120 REM LD A with green/black(&H21), Out(&HBF) A
130 REM LD A with register destination
140 REM Out(&HBF) A, Enable int's, Return
```

NOTE:
On return to BASIC, ie after pressing Break, you will notice that the screen reverts to black on green. This is because BASIC gets the color information from address &H9339.

WRITING TO VRAM:

Load the HL register with the screen address then call &H2C44, and output the value to port &HBE. The address is auto-incremented by one location after each write, eg,

```
ENTRY: A000  F3       D1           ; Disable Interrupts
       A001  D3BF     IN(BF),A     ; Clear Status register
BEGIN: A003  21003C   LD HL,3000   ; Text screen
       A006  CD442    CALL 2C44    ; Write address
       A009  0610     LD B,10      ; 16 times
       A00B  3E32     LD A,32      ; Character "2"
LOOP:  A00D  D3BE     OUT(BE),A    ; Print it
       A00F  10FC     DJNZ LOOP    ; 16 times
       A010  09       RET          ; Back to BASIC
```

READING FROM VRAM:

Load the HL register with the screen address, call &HC32, then input the value from port &HBE. The address is autoincremented after each read.

****REMEMBER****
Disable interrupts, then read the status register at port &HBF before you do what you want, or you will get strange results.

ALTERING THE CURSOR POSITION:

If using machine-code the above procedures dealing with reading/writing to video RAM are required to set up the 14 bit address pointer. However, if using a hybrid program, ie a mixture of machine-code and BASIC, especially when calling the print routine at &H4A6F, then the cursor position may be altered by poking the appropriate X and Y values into locations &H9489 and &H948A respectively before calling the print routine. An example of this is given in the next section.

WRITING TEXT OR CHARACTERS TO VRAM:

There is a routine in ROM which allows the user to move data to the Video RAM. The following program illustrates this. The text is hidden in the data statements, and a machine code subroutine is used to point the HL register to the text, then the ROM routine at &H4A6F is called. This writes the text out to Video RAM at the current cursor position. Note that the text must end in &HOD or 13 decimal, and you can also clear the screen etc, by the use of control codes (CLS=12 decimal).

```
 10 SCREEN 1,1
 20 FOR X=&HA000 TO &HA010 : REM the machine code
 30 READ A: POKE X,A NEXTX
 40 FOR Z=&HB000 TO &HB000 : REM the text string
 50 READ S: POKE Z,S: NEXT Z
 60 CALL &HA000
 70 REM Change cursor x,y positions
 80 DATA &H3E,&H0F,&H32,&H89,&H94
 90 DATA &H3E,&H0A,&H32,&H8A &H94
100 REM Machine-code routine
110 DATA &H21,&H00,&HB0,&HCD,&H6F,&H4A,&HC9
120 REM Text message follows
130 DATA 78,111,116,32,66,97,100,32,101
140 DATA 104,33,33,33,13
```

```
ENTRY:    A000     3E0F      LD,0F
          A002     328994    LD (9489),A ; X position = 15
          A005     3E0A      LD A,0A
          A007     328A94    LD (948A),A ; Y position = 10
          A00A     2100B0    LD HL,B000  ; Point to text
          A00D     CD6F4A    CALL 4A6F   ; Call print routine
          A010     C9        RET         ; Back to BASIC
TEXT:     B000     ""                    ; 'Not Bad eh!(0D)'
```

There is also another routine used for writing a string of characters to the video screen. This routine is at &H2400 and may be used in the following way,

```
 10 SCREEN 1,1: CLS
 20 FOR X=&HA000 TO &HA009
 30 READ A: POKE X,A: NEXT
 40 CALL &HA000
 50 STOP
 60 DATA &H3E,&H32,&H06,&H20
 70 DATA &HCD,&H00,&H24,&H20
 80 DATA &HFB,&HC9
 90 REM LD A with "2"
100 REM LD B with number of times to be printed
110 REM Call routine at &H2400
120 REM dec B and Jp not zero to print routine
130 REM Return when B is zero
```

SPRITES:

A sprite is a predefined graphic character. This can be one of four possible sizes, eight by eight pixels, sixteen by sixteen pixels (double the first), or thirty-two by thirty-two pixels (double the second). The sprite may be moved pixel by pixel around the screen, and a test may be made to see if any two sprites overlap by a single pixel element. The sprite size is controlled by the MAG command by SEGA BASIC, and the actual shape of the sprite is defined by the PATTERN command. The position of the sprite is controlled by the SPRITE command.

SPRITE PLANES:

The SEGA uses thirty-two planes, where each plane can be thought of as a transparent screen each behind the other. Only one sprite can be present on a sprite plane at any one time, but as the planes are stacked behind each other, sprites appearing on the closest plane have the highest display priority, ie, they appear in front if the sprites on the planes behind it. Sprites can thus appear to move in front of, or behind other sprites, depending on which planes are used.
The pattern plane, or the plane on which ordinary text is written to, is the lowest priority, thus sprites will always appear in front if written text.

SPRITE COLLISION DETECTION:

Sprite collisions may be detected by reading the Status register located at port &HBF. If any two sprites overlap by a single pixel, bit 5 will be set to logic 1. A basic program to test this would be:

```
10 SCREEN 2,2: CLS
20 PRINT "Sprite collision demo"
30 FOR DE=1 TO 1600: NEXT DE
40 PATTERNS#1,"FFFFFFFFFFFFFFFF"
50 PATTERNS#2,"FFFFFFFFFFFFFFFF"
60 SPRITE 0,(120,20),1,14: C=1
70 FOR X= 0 TO 255
80 B=INP(&HBF): IF (B AND 32)=32 THEN GOSUB 120
90 SPRITE 1,(X,20),2,C
100 IF INKEY$="" THEN GOTO 100
110 NEXT X: STOP
120 CURSOR 20,10: PRINT CHR$(5);"Collision"
130 BEEP: C=4: RETURN
```

Machine code programs may look something like,

```
A000      DBBF     INP(&HBF),A
A002      E620     AND 20
A004      FE20     CP  20
A006      28??     JR Z Collision
A008               Continue with main program
```

SPRITES ATTRIBUTES TABLE:

Starting at address &H3B00 are four bytes for each sprite. These groups of four bytes control the position, colour and number of each sprite. Sprite 0 has the first four locations, sprite 1 the second group of four bytes, etc. Refer to Table XXX7 for the relative locations. Table XX26 lists a machine code program which creates sprites, moves them to screen, checks for sprite collision, changes their colour, beeps, and gets a response from the keyboard &all using mcode!)

SPRITE PATTERN GENERATOR TABLE:

Located at address &H1800 are eight bytes for each sprite. These locations hold the pattern for the sprites, as defined by the BASIC command PATTERN. This area also contains the eight by eight patterns for the text screen. They are swapped over as needed by the routine at &H2BD4. The following BASIC program illustrates the creation of a sprite, and it's movement by poking the attribute area of the VRAM.

```
 10 SCREEN 2,2: CLS: PRINT" Sprite Demo"
 20 B=&H1800:REM Create the Sprite
 30 FOR X=0 TO 7: READ A
 40 POKE B+X,A: NEXT
 50 B=&H3B00: REM Create attributes
 60 FOR X=0 TO 3: READ A
 70 POKE B+X,A: NEXT
 80 FOR X=0 TO 255
 90 POKE &H3B01,X
100 NEXT X:POKE &H3B03,4
110 GOTO 110
120 DATA &HFF,&HFF,&HFF,&HFF,&HFF,&HFF,&HFF,&HFF
130 DATA 32,0,0,15
140 REM Y=32,X=0,SPRITE0,COLOR15
```

PATTERN GENERATOR TABLES:

These addresses store the eight bytes that are needed to compose the character. For the Text mode, the patterns are loaded from ROM address &H1000 into the VRAM area when the computer is turned on or reset.

ALTERING THE CONTENTS OF THE TEXT PATTERN GENERATOR TABLE:

In the text mode, the 8x8 patterns which make up the character are stored at address &H1800 onwards. Only the characters from &H20 to &HFF are defined in the pattern table, thus the pattern for each character is obtained by using the following formula,

address = &H1800 + character value*8

This gives the address of the first byte that makes up the character. The other seven bytes follow the address determined

by the formula. This information can now be used to alter the contents of the existing characters so as to provide both normal and inverse video characters on the text screen at the same time. Basically, the following program replaces the eng/diers characters with the equivalent inverse video alphanumeric set.

```
   10 SCREEN 1,1:AZ$="";FOR A=1 TO 14
   20 READ AS;AZ$+CHR$(AS):NEXT
   30 DATA &HA9,&HAE,&HB6,&HA5,&HB2,&HB3
   40 GOSUB 2000: CLS
   50 PRINT"  Welcome to  ";AZ$:PRINT
   60 PRINT"  Try printing out the  ";CHR$(&HC5);CHR$(&HCE);
      CHR$(&HC7);"/DIER'S"
   70 PRINT" characters.": PRINT
   80 STOP
2000 B+H1800+&H40*8:C=&H1800+&H7F*8
2010 DC=(C+&H20*8)+8
2020 FOR X=B TO C STEP 8
2030 FOR A=X TO X+7
2040 DA=VPEEK(A)
2050 DB=DA XOR &HFF
2060 VPOKE(DC),DB:DC=DC+1
2070 NEXT:NEXT:RETURN
```

By manipulating the contents of the pattern tables, it would be easy to create upside down and reverse characters as well. Table XX27 lists such a program.

NAME TABLE ADDRESSES:

These are eight bit pointers which point to the specific pattern required. If using the Text mode, it represents the ASCII equivalent of the character.

MULTI-COLOUR MODE:

Table XXX9 lists a program which experiments with the multi-colour screen mode. A machine code routine is poked into memory and when called, it switches over to the multi-colour mode. Be sure to try this program with a colour television set, as it is quite impressive. The colour attributes for the multi-colour mode are stored at &H3800 to &H3B00. Poking these areas with different values in the range 0 to 255 can result in very colourful displays.

SWAPPING THE CONTENTS OF THE TEXT SCREEN:

Utilising the large memory available with the 32K RAM cartridge, it is possible to create a screen swap routine. This involves reading the entire contents of the text screen into a buffer, and then carrying on as per normal. When the old screen is required, a routine is called which rewrites the buffer back to the screen. The following program illustrates this. A machine code routine is poked into line 5 of the program.

```
5 REM AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
 10 SCREEN 1,1:CLS:PRINT" Text Screen Swap"
 20 FOR X=&H9808 TO &H983B
 30 READ A!POKE X,A:NEXT
 40 PRINT" This is the original screen."
 50 FOR DE=1 TO 350: NEXT DE:CALL &H9808
 60 CLS:PRINT" This is the new screen."
 70 FOR DE=1 to 350:NEXT DE
 80 CALL &H9822
 90 PRINT" How's that!"
100 STOP
110 DATA &HF3,&HDB,&HBF,&H21,&H00,&H3C,&HCD,&H32,&H2C,&HF3,
    &H21,&H00,&HA0,&H06,&H05,&HC5,6,193,&H0E,&HBE,&HED,&HB2,
    &HC1,&H10,&HF6,&HC9
120 DATA &HF3,&HDB,&HBF,&H21,&H00,&H3C,&HCD,&H44,&H2C,&HF3,
    &H21,&H00,&HA0,&H06,&H05,6,193,&H0E,&HBE,&HED,&HB3,&HC1
    &H10,&HF6,&HC9
```

The routine at &H9809 saves the text screen contents into the main RAM starting at location &HA000 onwards, while the routine at &H9822 writes the buffer at location &HA000 to VRAM. Refinement of this could result in simple animation. In machine code the program is.

```
9803    F3      DI          ; Disables interrupts
9809    DBBF    IN A,(BF)   ; Clear status register
980B    21003c  LD HL,3000  ; Text screen address
980E    CD322C  CALL 2C32   ; Set up VDP for read
9811    FE      DI
9812    2100A0  LD HL,A000  ; Buffer area
9815    0605    LD B,05
9817    C5      PUSH BC     ; Read
9818    06C0    LD B,C0
981A    0EBE    LD C,BE     ; C= Port BE
981C    EDB2    INIR        ; Read until B=0
981E    C1      POP BC
981F    10F6    DJNZ Read   ; Complete screen?
9821    C9      RET
9822    F3      DI
9823    DBBF    IN A,(BF)   ; Clear status register
9825    21003C  LD HL,3000  ; Text screen address
9828    CD442C  CALL 2C44   ; Set up VDP for write
982B    F3      DI
982C    2100A0  LD HL,A000  ; Buffer area
982F    0605    LD B,05
9831    C5      PUSH BC     ; Write
9832    06C0    LD C,C0
9834    0EBE    LD C,BE     ; C = VDP
9836    EDB2    OUTR        ; Do until B=0
9838    C1      POP BC
9839    10F6    DJNZ Write  ; All the screen
983B    C9      RET
```

Table XXX5: MODE SELECT BITS.

MS1	MS2	MS3	Screen type
0	0	0	Graphics mode I (32 x 24)
0	0	1	Graphics mode II (256 x 192)
0	1	0	Multicolour mode (64 x 48)
1	0	0	Text mode (40 x 24)

Table XXX6: SIZE & MAG BITS.

Mag	Size	Bit size	SEGA Manual
0	0	8 x 8	MAG 0 (single sprite)
0	1	16 x 16	MAG 1 (single sprite)
1	0	16 x 16	MAG 2 (double mag0)
1	1	32 x 32	MAG 3 (double mag1)

Table XXX7: SPRITE ATTRIBUTE TABLE.

Y POSITION	EC If logic one, it shifts the sprites to the left by 32 pixels.
X POSITION	
SPRITE NAME	COLOUR The 4 bits make up the colour of the sprite. Refer to Table XXX8 for the colour values.
EC1010101COLOUR	

Table XXX8: COLOUR VALUES.

0	Transparent	8	Red
1	Black	9	Light Red
2	Green	A	Deep Yellow
3	Light Green	B	Light Yellow
4	Dark Blue	C	Dark Green
5	Light Blue	D	Magenta
6	Dark Red	E	Gray
7	Cyan	F	White

Table XXX9: MULTI-COLOUR MODE PROGRAM.

```
 5 DEFFNA(R) = INT(RND(1)*R) + &H3800
10 SCREEN 2,2 : CLS
20 FOR X = &HA000 TO &HA011
30 READ A : POKE X,A : NEXT A
40 DATA &HF3,&H3E,&H00,&HD3,&HBF
50 DATA &H3E,&H80,&HD3,&HBF
60 DATA &H3E,&HC8,&HD3,&HBF
70 DATA &H3E,&H84,&HD3,&HBF,&HC9
80 DH=&H11: DF=&H3800: DG=&H3B00
90 FOR DE=DF TO DG: VPOKE DE,DH
100 NEXT
110 CALL &HA000
120 X = FNA(&H300)
130 VPOKE X,RND(1)*&HFF
140 GOTO 120
```

In Machine Code

```
A000    F3      DI          ; Disables interrupts
A001    3E00    LD A,00     ; Select multi-mode
A003    D3BF    OUT (BF),A
A005    3E80    LD A,80     ; Register 0
A007    D3BF    OUT (BF),A
A009    3EC8    LD A,C8     ; Multi-mode
A00B    D3BF    OUT (BF),A
A00D    3E84    LD A,84     ; Register 1
A00F    D3BF    OUT (BF),A
A011    C9      RET
```

SAVE/LOAD/VERIFY BASIC EXTENSION

By MICHAEL HADRUP (C) 1986 MJH Software

This is the first in a new series of machine code programs that extend the Basic interpreter of the Level IIIB cartridge and Disk Basic. Once the commands are loaded and initialised by a machine code CALL, they continue to function until power off, or BOOT (Disk Basic) is executed.

The MC routines for these commands are stored in high memory, at &HF900 to &HFFFF, and are initialised by CALL &HFB67. These routines add new commands that allow the Loading/Saving/Verifying of Basic programs with variables, videoram (text screen, sprite data etc), graphics screen pictures, and bytes of MC and data. Also the commands can vector any errors and the break key.

Now to the typing . . .

LSV FOR DISK BASIC

Type in listing 1 carefully. Each line of data contains 32 bytes of machine code followed by a checksum for that line. The program can be saved at any time by typing SAVE "LSV.Dta"

When finished save it, type RUN and wait. The MC is being poked to &HF900. If all goes well then Basic will return the message "No errors!!". If you dont get this message then check the data line for typing errors, and type RUN again.

When you get (finally?), the message "No errors!!", insert a disk on which the code can be saved and press Space. Next save the program using SAVE "LSV.Dta".

Type NEW, then enter listing 2. This is the program that initialises LSV. Save this using SAVE "LSV".

Now type RUN and skip the section on cartridge Basic, and look at using the extended commands.

LSV FOR CARTRIDGE IIIB BASIC

Start by typing in a "1 REM" followed by more than 256 characters. (This is easiest done, by typing in "1 REM" followed by 7 lines of "O" 's.) When listed Basic has automatically truncated the line to the required amount of bytes and your program should look like this . . .

```
1 REM OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
```

Type PRINT PEEK (&H9800). If the answer is not 251, then add some more characters to the REM statement and recheck.

Now duplicate line 1 and line 2. Duplicate line 2 as line 3 and so on, until you have 8 lines of the same REM statement, numbered 1 to 8. To make the listing easier to read, type this as a direct command.

```
FOR N=0T07:POKE&H9806+N*257,13:NEXT
```

Now add to this program, listing 3. Each line of data contains 32 bytes of machine code, followed by a checksum for that line. The program can be saved at any time by typing SAVE "LSV.Dta"

When finished, save it, type RUN and wait. The MC is being poked in to the REM statements. If all goes well then Basic will return the message "No errors!!". If you don't get this message then check the data line for typing errors, and type RUN again.

When you get (finally?), the message "No errors!!", save the program using SAVE "LSV.Dta"

Now delete lines 100 onwards by typing DELETE 100-, and add to the program listing 4. This is the program that copies LSV to &HF900, and initialises the commands. Save this by typing as a direct command . . .

```
A$=CHR$(7)+CHR$(7)+CHR$(7)+CHR$(7)+CHR
$(7)+CHR$(12)+"LSV"+CHR$(10)+CHR$(10)+
CHR$(13)+"RUN"+CHR$(11):POKE&H82A2,16:
FORN=1TO16:POKE&H82A2+N,ASC(MID$(A$,N,
1)):NEXT:CALL&H7A40
```

USING THE EXTENDED COMMANDS

All the extended commands are prefixed by an "*" to show they are extended commands. The extended commands and syntax are as follows . . .

All Parameters on *LOAD and *VERIFY are optional. File name defaults to first file found of the same type.

*LOAD and *SAVE are replaced within commands as *CLOAD and *CSAVE in Disk Basic.

*LOAD "filename"
Loads from tape a Basic program with variables, and autoruns if a line no was specified or returns to Basic and prints bytes free.

*LOADS "filename"
Loads from tape a graphics screen picture.

*LOADV "filename", start
Loads from tape video ram data. If start is not specified then loads to original address.

*LOADC "filename", start
Loads from tape RAM/ROM data. If start is not specified then loads to original address.

*SAVE "filename"
Saves to tape a Basic program with variables.

*SAVE "filename", RUN lineno
Saves to tape a Basic program with variables, that autoruns at the line specified when loaded.

*SAVES "filename"
Saves to tape a graphics screen picture. The is saved from &H0000 to &H17FF (bit map) and &H2000 to &H37FF (colour), in video ram.

*SAVEV "filename", start, end
Saves to tape video ram data from start to end.

*SAVEC "filename", start, end
Saves to tape RAM/ROM data from start to end.

If CHR$(0) is used as the first byte of a filename during saving, any attempt to break during loading, or if a tape loading error occurs, this results in a reset of the computer or a BOOT (Disk Basic)

For verifying programs, *LOAD and *CLOAD are replaced by *VERIFY, with all parameters identical.

*ON B action
When the break key is pressed this vectors it to a required action.

*ON E action
When an error occurs this vectors it to a required action.

The possible actions are . . .
GOTO lineno
GOTO to line specified.

RUN lineno
RUN line specified.

RUN
RUN program.

NEW
NEW program.

ERASE
This resets the computer, or BOOTS (Disk Basic).

CONT
The break/error is ignored.

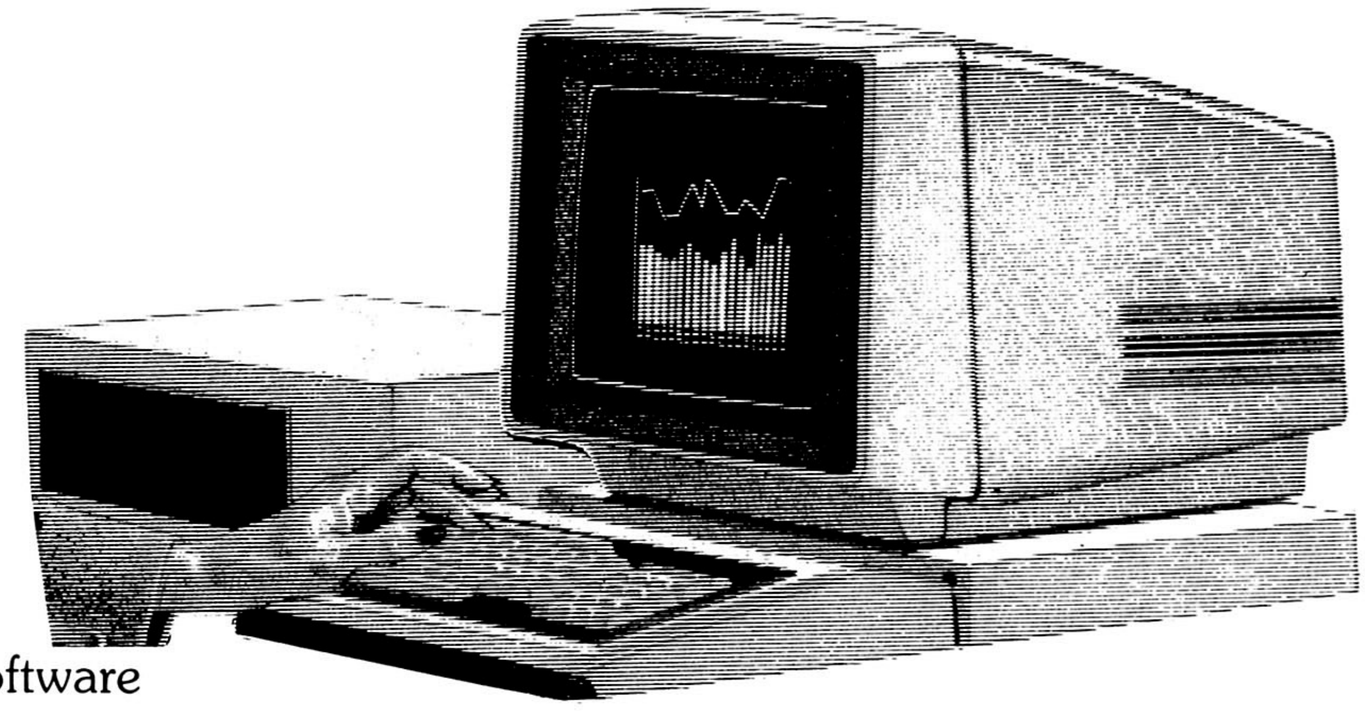
STOP
The break/error is reset to normal.

ALPHABETICAL SORTING

By C. Plows 1986

```
100 REM ALPHABETICAL SORTING
110 REM By C. Plows 1986
120 REM
130 DIM W$(100)
140 CLS
150 INPUT "HOW MANY WORDS =";A
160 CLS
170 FOR I=1 TO A
180 PRINT I;:INPUT W$(I)
190 NEXT I
200 C=A-1
210 FOR I=1 TO C
220 B=I
230 IF W$(B)<=W$(B+1)THEN 290
240 B$=W$(B)
250 W$(B)=W$(B+1)
260 W$(B+1)=B$
270 B=B-1
280 IF B<70 THEN 230
290 NEXT I
300 CLS
310 FOR I=1 TO A
320 PRINT I;")";W$(I)
330 NEXT I
340 END
```

PRINTER SHADING



(C) 1986 MJH Software

This program shades colour pictures on the graphics screen and dumps them to a dot matrix printer connected to the Centronics printer port.

To use the program type in the listing and save using SAVE "PSHADEV.Dta"

RUN the program, and save the code using SAVEM "PSHADEV.Cde",&HF600,&Hf7ff

To use the program display the picture on the graphics screen, and type CALL &HF65B

At the moment I'm working on versions for the SP-400 and FS232 ports. Here are some examples of pictures Loaded from a ZX SPECTRUM, into the SEGA and dumped to a printer.

```
1 REM PRINTER SHADE
2 REM
3 REM By Michael Hadrup
4 REM
5 REM (C) 1986 MJH Software
6 REM
7 REM
10 RESTORE
20 X=&HF600
30 FORN=1000TO1400STEP10
40 C=0
50 FORM=0TO7
60 READA$
70 POKEX,VAL("&H"+A$)
80 C=C+PEEK(X)
90 X=X+1
100 NEXT
110 READA$
120 IFC<>VAL("&H"+A$)THENBEEP2:PRINT"Error in line ";N:STOP
130 NEXT
140 BEEP
150 BEEP
160 PRINT"No errors!!"
170 END
1000 DATA 10,10,2E,3D,15,29,1A,47,12A
1010 DATA 33,38,42,4C,1F,24,51,56,1E3
1020 DATA F0,F0,F0,F0,F0,C0,70,F0,6D0
1030 DATA E0,30,C0,70,E0,E0,30,C0,4F0
1040 DATA 70,54,E0,30,C0,70,54,E0,438
1050 DATA A,80,70,54,E0,A,80,70,328
1060 DATA 20,E0,A,80,70,0,E0,A,2E4
1070 DATA 80,30,0,C0,10,80,20,0,220
1080 DATA C0,10,80,20,0,40,10,80,240
1090 DATA 20,0,40,0,0,20,0,40,C0
1100 DATA 0,0,0,0,40,0,0,0,40
1110 DATA 0,0,0,F3,CD,F6,F6,1E,3CA
1120 DATA 31,CD,EE,F6,21,0,0,1,304
1130 DATA 80,60,1E,4C,CD,EE,F6,AF,4AA
1140 DATA CD,FD,F6,3E,5,CD,FD,F6,5C3
1150 DATA 7D,D3,BF,7C,D3,BF,0,0,41D
1160 DATA 0,0,DB,BE,32,42,F7,0,304
1170 DATA 0,0,0,DB,BE,32,44,F7,306
1180 DATA 7D,D3,BF,7C,C6,20,D3,BF,503
1190 DATA 0,0,0,0,DB,BE,32,43,20E
1200 DATA F7,0,0,0,0,DB,BE,32,2C2
1210 DATA 45,F7,79,D9,4F,21,42,F7,437
1220 DATA CD,25,F7,D5,CD,25,F7,E1,588
1230 DATA 6,5,1A,7,7,7,7,B6,F7
1240 DATA CD,FD,F6,1C,2C,10,F3,D9,4E4
1250 DATA CB,9,30,DE,7D,C6,8,6F,39C
1260 DATA 30,A6,3E,6,D3,DE,DB,DD,483
1270 DATA 1F,D2,C2,7,CD,F6,F6,7D,4F0
1280 DATA C6,2,E6,7,6F,20,1,24,269
1290 DATA 5,C2,6A,F6,FB,C9,3E,1B,444
1300 DATA CD,FD,F6,7B,18,7,3E,A,3A2
1310 DATA CD,FD,F6,3E,D,8,C5,E5,4BD
1320 DATA 2E,5,1,0,0,DB,E4,E6,2D9
1330 DATA 2,28,D,B,78,B1,20,F5,280
1340 DATA 2D,20,EF,3E,40,C3,D3,75,3C5
1350 DATA 8,D3,E5,3E,E,D3,E7,3C,402
1360 DATA D3,E7,E1,C1,C9,7E,A1,8,54C
1370 DATA 23,7E,23,8,28,6,8,F,111
1380 DATA F,F,F,8,8,E6,F,20,152
1390 DATA 3,3A,E1,AB,16,F6,5F,1A,34E
1400 DATA 5F,C9,0,0,0,0,18,1,141
```

LISTING 1
DATA FOR DISC BASIC VERSION

```
10 CLS
20 X=&HF900:RESTORE1000
30 FORN=1000TO2000STEP20:C=0
40 CURSOR0,0:PRINTHEX$(X)
50 FORM=0TO31:READA$:POKEX,VAL("&H"+A$):C=C+PEEK(X):X=X+1:NEXTM
60 READA$:IFC<>VAL("&H"+A$)THENBEEP2:PRINT"Error in line ";N:STOP
70 NEXT
80 BEEP:BEEP:PRINT"No errors!!"
90 PRINT:PRINT"Insert DISK and press space"
100 A$=INKEY$:IFA$<>" "THEN100
110 SAVEM"LSV Dsk.Cde",&HF900,-1
120 PRINT:PRINT"Now SAVE Basic program
130 END
1000 DATA CD,1E,FA,21,F2,F9,E5,14,8,15,DB,DD,1F,E6,40,F6,8,4F,BF,C0,CD,C7,F9,3E,
4F,D3,7F,30,F6,21,15,4,109C
1010 REM
1020 DATA 10,FE,2B,7C,B5,20,F9,CD,C3,F9,30,E7,6,9C,CD,C3,F9,30,E0,3E,C6,B8,30,DC
,24,20,F1,6,C9,CD,C7,F9,12B7
1030 REM
1040 DATA 30,D1,78,FE,D4,30,F4,CD,C7,F9,D0,79,E6,40,F6,4,4F,26,0,6,B0,18,52,8,20
,20,30,28,8,DD,CB,0,E4A
1050 REM
1060 DATA 7E,28,11,FD,E5,D9,E1,7D,D3,BF,7C,D3,BF,D9,7D,D3,BE,8,18,2F,FD,75,0,8,1
8,29,CB,11,AD,C0,79,1F,1042
1070 REM
1080 DATA 4F,13,18,21,8,DD,CB,0,7E,28,12,FD,E5,D9,E1,7D,D3,BF,7C,D3,BF,D9,0,0,8,
DB,BE,18,4,8,FD,7E,ED5
1090 REM
1100 DATA 0,AD,C0,FD,23,1B,8,6,B2,2E,1,CD,C3,F9,D0,3E,CB,B8,CB,15,6,B0,30,F3,7C,
AD,67,7A,B3,20,98,7C,F5B
1110 REM
1120 DATA FE,1,C9,CD,C7,F9,D0,3E,16,3D,20,FD,A7,4,C8,DB,DD,1F,D0,A9,E6,40,28,F5,
79,2F,4F,E6,F,F6,10,D3,119E
1130 REM
1140 DATA BF,3E,87,D3,BF,79,E6,40,3E,90,20,2,F6,A,D3,7F,37,C9,FB,F5,3A,16,AB,3D,
3A,19,AB,28,3,3A,E1,AB,F0E
1150 REM
1160 DATA D3,BF,3E,87,D3,BF,CD,2B,FA,6,14,DB,DD,1F,30,4,10,F9,F1,C9,3A,1,FB,A7,C
A,1B,5,C3,92,21,8,F3,FFB
1170 REM
1180 DATA DD,21,E4,FA,3E,84,D3,7F,AF,D3,7F,3E,9F,D3,7F,3E,92,D3,DF,3E,6,D3,DE,8,
C9,CD,1E,FA,21,F2,F9,E5,133E
1190 REM
1200 DATA 21,0,15,CB,7F,28,3,21,0,8,8,13,FD,2B,E,8,10,FE,79,E6,10,D3,DE,CD,D7,FA
,6,9A,2D,20,F1,5,BDC
1210 REM
1220 DATA 25,F2,50,FA,3E,6,6,2F,10,FE,D3,DE,3E,16,6,37,10,FE,D3,DE,6,35,8,6F,18,
26,7A,B3,28,2A,8,DD,C3D
1230 REM
1240 DATA CB,0,7E,28,15,FD,E5,D9,E1,7D,D3,BF,7C,D3,BF,D9,0,0,DB,BE,6F,5,5,8,18,4
,FD,6E,0,8,7C,AD,EEA
1250 REM
1260 DATA 67,E,14,3E,6,37,18,1D,6C,18,F3,3E,16,CB,78,10,FE,30,4,6,3C,10,FE,D3,DE
,8,CD,D7,FA,8,6,38,B76
1270 REM
1280 DATA 20,E9,5,EE,10,CB,15,20,E6,1B,FD,23,6,2B,DB,DD,1F,D0,7A,3C,20,A4,C9,79,
EE,1F,4F,F6,10,D3,BF,3E,EF3
1290 REM
1300 DATA 87,D3,BF,C9,0,0,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,0,0,0,
0,0,0,0,0,0,0,0,4E2
1310 REM
1320 DATA 0,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,0,0,0,0,0,0,0,0,0,0,
0,50,72,6F,67,398
1330 REM
1340 DATA 72,61,6D,0,42,79,74,65,73,0,56,69,64,65,6F,20,72,61,6D,0,53,63,72,65,6
5,6E,24,0,53,74,61,72,ABC
```

```
1350 REM
1360 DATA 74,20,74,61,70,65,2C,20,74,68,65,6E,20,70,72,65,73,73,20,61,6E,79,20,6
B,65,79,D,0,0,0,8E,FB,AED
1370 REM
1380 DATA 48,7,0,0,0,0,1,21,A1,FB,22,D5,9E,21,80,FB,22,5B,AD,21,8E,FB,22,5E,FB,2
1,48,7,22,60,FB,C9,C43
1390 REM
1400 DATA 31,10,A3,CD,3E,38,2A,5E,FB,E5,2A,62,FB,C9,AF,32,CD,AE,C3,C2,7,CD,3C,62
,C3,BD,3,3A,66,FB,C3,A5,10B8
1410 REM
1420 DATA 1A,32,66,FB,FE,1F,20,8,8,1A,13,FE,C1,28,F,8,31,10,A3,CD,3E,38,2A,60,FB
,E5,2A,64,FB,C9,CD,3E,D13
1430 REM
1440 DATA 84,13,FE,A0,20,70,CD,3E,84,FE,42,28,4,FE,45,20,DE,F5,CD,3D,84,13,FE,86
,28,2F,1,95,FB,FE,8B,28,FB4
1450 REM
1460 DATA 39,1,1B,5,FE,A2,28,32,1,59,46,FE,9D,28,25,1,E0,5,FE,87,28,24,FE,97,C2,
23,FD,F1,F5,1,8E,FB,E7A
1470 REM
1480 DATA FE,42,28,16,1,48,7,18,11,CD,16,FD,1,D3,5,28,9,1,8F,5,CD,F2,83,DA,23,FD
,CD,1F,FD,F1,FE,45,DCF
1490 REM
1500 DATA 28,A,ED,43,5E,FB,22,62,FB,C3,11,FD,ED,43,60,FB,22,64,FB,C3,11,FD,21,8A
,88,CD,E1,83,DA,AF,FB,4F,121F
1510 REM
1520 DATA 1A,FE,43,6,C,28,C,FE,56,6,8,28,6,FE,53,20,6,6,4,13,79,80,4F,79,D6,88,8
,AF,32,8F,9A,3C,A32
1530 REM
1540 DATA 32,1,FB,CD,16,FD,28,1C,13,FE,2C,28,17,1B,CD,38,68,DA,23,FD,CD,16,FD,28
,6,FE,2C,C2,23,FD,13,3E,DBB
1550 REM
1560 DATA 1,CD,EB,74,3A,8F,9A,CD,2C,15,21,90,9A,A7,20,A,8,CD,8E,FE,CA,23,FD,8,36
,FF,D5,11,E6,FA,1,10,F19
1570 REM
1580 DATA 0,ED,B0,D1,8,FE,8,38,4C,8,21,0,0,1,FF,FF,CD,16,FD,28,2E,FE,2C,28,B,CD,
FE,4A,CD,BA,FE,44,E99
1590 REM
1600 DATA 4D,21,0,0,8,CD,8E,FE,20,1F,8,CD,3E,84,FE,2C,20,51,13,CD,FE,4A,CD,BA,FE
,A7,ED,42,38,45,28,43,DAB
1610 REM
1620 DATA 23,18,7,8,CD,8E,FE,28,3A,8,ED,43,F8,FA,22,F6,FA,CD,1F,FD,8,21,D,FD,E5,
ED,53,5C,FB,4F,CB,3F,1032
1630 REM
1640 DATA CB,3F,CD,61,70,2,FE,28,FD,4D,FD,59,FD,ED,5B,5C,FB,1A,13,C3,16,7,CD,3E,
84,FE,D,C8,FE,3A,C9,CD,1149
1650 REM
1660 DATA 16,FD,C8,3E,23,C3,A1,FB,CD,1F,FD,3E,3,32,E5,FA,21,0,18,22,F6,FA,22,FA,
FA,EB,21,0,20,22,FC,FA,107B
1670 REM
1680 DATA 21,0,0,22,F8,FA,79,FE,5,28,6C,18,16,3E,2,32,E5,FA,79,FE,9,28,5D,18,A,3
E,1,32,E5,FA,79,FE,CB2
1690 REM
1700 DATA D,28,51,CD,CD,FE,ED,5B,11,FB,2A,F8,FA,7C,FE,FF,20,6,BD,20,3,2A,13,FB,C
D,A8,FE,E5,FD,E1,CD,9C,12E4
1710 REM
1720 DATA FE,28,1,37,3E,FF,F5,C5,CD,0,F9,30,1B,C1,F1,8,3A,E5,FA,FE,3,C0,2A,FC,FA
,ED,5B,FA,FA,CD,AE,FE,13CF
1730 REM
1740 DATA E5,FD,E1,8,CD,0,F9,D8,3A,1,FB,A7,CA,1B,5,3E,41,C3,A1,FB,2A,F8,FA,E5,21
,3C,FB,CD,17,4B,CD,6A,11CD
1750 REM
1760 DATA 56,A7,28,FA,AF,32,E4,FA,FD,21,E5,FA,11,1B,0,AF,CD,39,FA,6,32,76,10,FD,
FD,E1,3A,E5,FA,FE,2,3E,11A6
1770 REM
1780 DATA FF,38,3,32,E4,FA,ED,5B,F6,FA,F5,CD,39,FA,F1,8,3A,E5,FA,FE,3,C0,FD,2A,F
C,FA,ED,5B,FA,FA,8,C3,156E
1790 REM
1800 DATA 39,FA,AF,32,E5,FA,CD,16,FD,21,0,0,28,18,CD,3E,84,FE,86,C2,23,FD,13,79,
3D,C2,23,FD,CD,FE,4A,CD,10B6
1810 REM
```

```
1820 DATA 1F,FD,ED,53,5C,FB,22,FE,FA,79,3D,28,41,CD,CD,FE,2A,54,99,ED,5B,11,FB,7
9,A7,C2,7B,FD,E5,19,22,5A,11C3
1830 REM
1840 DATA 99,D5,ED,5B,5C,99,1B,ED,52,3E,1C,D2,A1,FB,D1,C1,2A,13,FB,9,22,56,99,2A
,15,FB,9,22,58,99,60,69,EDO
1850 REM
1860 DATA CD,7B,FD,2A,19,FB,7C,B5,CA,BD,3,C3,5D,46,2A,5A,99,ED,5B,54,99,D5,ED,52
,22,F6,FA,2A,56,99,ED,52,1174
1870 REM
1880 DATA 22,F8,FA,2A,58,99,ED,52,22,FA,FA,C3,B8,FD,FE,D,C8,FE,B,DO,C5,4F,E6,3,3
D,79,C1,C9,79,FE,E,C8,1332
1890 REM
1900 DATA FE,8,DO,E6,3,FE,2,C9,3A,E5,FA,FE,2,D8,CD,C6,FE,CD,9C,FE,C8,7C,F6,40,67
,C9,8,F5,E6,C,FE,8,1413
1910 REM
1920 DATA CC,C6,FE,F1,8,C9,7C,E6,CO,C2,23,FD,C9,AF,32,E4,FA,C5,FD,21,C,FB,FD,E5,
11,1B,0,AF,37,CD,0,F9,1376
1930 REM
1940 DATA FD,E1,30,F2,3A,0,FB,FE,4,30,EB,1,2,1,1E,FO,B8,28,3,B9,20,A,3A,E5,FA,B8
,28,B,B9,28,8,78,D8F
1950 REM
1960 DATA FD,BE,E5,28,2,1E,80,FD,46,0,4,21,1B,FB,4B,AF,ED,B1,10,FA,CD,17,4B,3E,3
A,CD,B6,7B,4B,FD,E5,D1,102B
1970 REM
1980 DATA 21,E6,FF,19,6,10,7E,3C,20,3,79,80,4F,13,1A,BE,23,20,1,C,CD,B6,7B,10,F4
,3E,A,CD,B6,7B,3E,D,B28
1990 REM
2000 DATA CD,B6,7B,CB,79,20,8F,C1,3A,E5,FA,FE,2,D8,3E,FF,32,E4,FA,C9,0,0,0,0,0,0
,0,0,0,0,0,0,0,CB9
2010 REM
```

LISTING 2

INITIALISE ROUTINE FOR DISC BASIC VERSION

```
10 CLS:BEEP
20 PRINT"LOADING:
30 PRINT
40 PRINT"LOAD-SAVE-VERIFY Extension
41 PRINT:PRINT"(C) 1986 BY MJH SOFTWARE
50 LOADM"LSV Dsk.Cde",&HF900
60 LIMIT&HF900
70 CALL&HFB67
80 PRINT:PRINT
90 BEEP:BEEP
100 PRINT"LSV is installed in BASIC interpreter
110 PRINT:PRINT
120 PRINT"Bytes free=";FRE+(PEEK(&H995A)+256*PEEK(&H995B))-(PEEK(&H9954)+256*PEE
K(&H9955))
150 PRINT
230 CALL&H623C:REM NEW
```

LISTING 4

INITIALISE ROUTINE FOR CARTRIDGE IIIB BASIC

```
10 CLS:BEEP
20 PRINT"LOAD-SAVE-VERIFY Extension
30 PRINT:PRINT"(C) 1986 BY MJH SOFTWARE
40 POKE&H8168,0:POKE&H8169,&HF9
50 PRINT:PRINT
60 PRINT"Bytes free=";FRE+(PEEK(&H8166)+256*PEEK(&H6167))-(PEEK(&H8160)+256*PEEK
(&H8161))
70 CALL&H9F4E
```

LISTING 3
DATA FOR CARTRIDGE BASIC IIIB VERSION

```
100 CLS
110 X=&H9807:RESTORE1000
120 FORN=1000TO1970STEP140
130 FORF=0TO6:C=0
140 CURSOR0,0:PRINTHEX$(X)
150 FORM=0TO31:READA$:POKEX,VAL("&H"+A$):C=C+PEEK(X):X=X+1:NEXTM
160 READA$:IFC<>VAL("&H"+A$)THENBEEP2:PRINT"Error in line ";N+F*20:STOP
170 NEXTF:X=X+33
180 NEXTN
190 FORF=0TO2:C=0
200 CURSOR0,0:PRINTHEX$(X)
210 FORM=0TO31:READA$:POKEX,VAL("&H"+A$):C=C+PEEK(X):X=X+1:NEXTM
220 READA$:IFC<>VAL("&H"+A$)THENBEEP2:PRINT"Error in line ";1980+F*20:STOP
230 NEXTF
240 BEEP:BEEP:PRINT"No errors!!"
250 END
1000 DATA CD,1E,FA,21,F2,F9,E5,14,8,15,DB,DD,1F,E6,40,F6,8,4F,BF,C0,CD,C7,F9,3E,
4F,D3,7F,30,F6,21,15,4,109C
1010 REM
1020 DATA 10,FE,2B,7C,B5,20,F9,CD,C3,F9,30,E7,6,9C,CD,C3,F9,30,E0,3E,C6,B8,30,DC
,24,20,F1,6,C9,CD,C7,F9,12B7
1030 REM
1040 DATA 30,D1,78,FE,D4,30,F4,CD,C7,F9,D0,79,E6,40,F6,4,4F,26,0,6,B0,18,52,8,20
,20,30,28,8,DD,CB,0,E4A
1050 REM
1060 DATA 7E,28,11,FD,E5,D9,E1,7D,D3,BF,7C,D3,BF,D9,7D,D3,BE,8,18,2F,FD,75,0,8,1
8,29,CB,11,AD,C0,79,1F,1042
1070 REM
1080 DATA 4F,13,18,21,8,DD,CB,0,7E,28,12,FD,E5,D9,E1,7D,D3,BF,7C,D3,BF,D9,0,0,8,
DB,BE,18,4,8,FD,7E,ED5
1090 REM
1100 DATA 0,AD,C0,FD,23,1B,8,6,B2,2E,1,CD,C3,F9,D0,3E,CB,B8,CB,15,6,B0,30,F3,7C,
AD,67,7A,B3,20,98,7C,F5B
1110 REM
1120 DATA FE,1,C9,CD,C7,F9,D0,3E,16,3D,20,FD,A7,4,C8,DB,DD,1F,D0,A9,E6,40,28,F5,
79,2F,4F,E6,F,F6,10,D3,119E
1130 REM
1140 DATA BF,3E,87,D3,BF,79,E6,40,3E,90,20,2,F6,A,D3,7F,37,C9,FB,F5,3A,36,93,3D,
3A,39,93,28,3,3A,10,94,E36
1150 REM
1160 DATA D3,BF,3E,87,D3,BF,CD,2B,FA,6,14,DB,DD,1F,30,4,10,F9,F1,C9,3A,1,FB,A7,C
A,0,0,C3,FA,4B,8,F3,106D
1170 REM
1180 DATA DD,21,E4,FA,3E,84,D3,7F,AF,D3,7F,3E,9F,D3,7F,3E,92,D3,DF,3E,6,D3,DE,8,
C9,CD,1E,FA,21,F2,F9,E5,133E
1190 REM
1200 DATA 21,0,15,CB,7F,28,3,21,0,8,8,13,FD,2B,E,8,10,FE,79,E6,10,D3,DE,CD,D7,FA
,6,9A,2D,20,F1,5,BDC
1210 REM
1220 DATA 25,F2,50,FA,3E,6,6,2F,10,FE,D3,DE,3E,16,6,37,10,FE,D3,DE,6,35,8,6F,18,
26,7A,B3,28,2A,8,DD,C3D
1230 REM
1240 DATA CB,0,7E,28,15,FD,E5,D9,E1,7D,D3,BF,7C,D3,BF,D9,0,0,DB,BE,6F,5,5,8,18,4
,FD,6E,0,8,7C,AD,EEA
1250 REM
1260 DATA 67,E,14,3E,6,37,18,1D,6C,18,F3,3E,16,CB,78,10,FE,30,4,6,3C,10,FE,D3,DE
,8,CD,D7,FA,8,6,38,B76
1270 REM
1280 DATA 20,E9,5,EE,10,CB,15,20,E6,1B,FD,23,6,2B,DB,DD,1F,D0,7A,3C,20,A4,C9,79,
EE,1F,4F,F6,10,D3,BF,3E,EF3
1290 REM
1300 DATA 87,D3,BF,C9,0,0.20,20,20,20.20,20,20,20,20,20,20,20,20,20,20,20,0,0,0,
0,0,0,0,0,0,0.0,4E2
1310 REM
1320 DATA 0,20,20.20,20.20,20,20.20,20,20,20.20,20,20.20.20,20,0,0,0,0,0,0,0,0,0,
0,50,72,6F.67,398
```

```
1330 REM
1340 DATA 72,61,6D,0,42,79,74,65,73,0,56,69,64,65,6F,20,72,61,6D,0,53,63,72,65,6
5,6E,24,0,53,74,61,72,ABC
1350 REM
1360 DATA 74,20,74,61,70,65,2C,20,74,68,65,6E,20,70,72,65,73,73,20,61,6E,79,20,6
B,65,79,D,0,0,0,8E,FB,AED
1370 REM
1380 DATA 81,6D,0,0,0,0,1,21,A1,FB,22,E8,86,21,80,FB,22,A4,95,21,8E,FB,22,5E,FB,
21,81,6D,22,60,FB,C9,DAD
1390 REM
1400 DATA 31,30,8B,0,0,0,2A,5E,FB,E5,2A,62,FB,C9,AF,32,14,97,C3,E6,6D,CD,26,3D,C
3,B5,6A,3A,66,FB,C3,B7,F6D
1410 REM
1420 DATA 73,32,66,FB,FE,1F,20,8,8,1A,13,FE,C1,28,F,8,31,30,8B,0,0,0,2A,60,FB,E5
,2A,64,FB,C9,CD,10,C03
1430 REM
1440 DATA 23,13,FE,A0,20,70,CD,10,23,FE,42,28,4,FE,45,20,DE,F5,CD,F,23,13,FE,86,
28,2F,1,95,FB,FE,8B,28,E35
1450 REM
1460 DATA 39,1,0,0,FE,A2,28,32,1,88,67,FE,9D,28,25,1,43,6C,FE,87,28,24,FE,97,C2,
23,FD,F1,F5,1,8E,FB,E74
1470 REM
1480 DATA FE,42,28,16,1,81,6D,18,11,CD,16,FD,1,37,6C,28,9,1,F1,6B,CD,C4,22,DA,23
,FD,CD,1F,FD,F1,FE,45,E72
1490 REM
1500 DATA 28,A,ED,43,5E,FB,22,62,FB,C3,11,FD,ED,43,60,FB,22,64,FB,C3,11,FD,21,8A
,88,CD,B3,22,DA,AF,FB,4F,1190
1510 REM
1520 DATA 1A,FE,43,6,C,28,C,FE,53,6,4,28,6,FE,56,20,6,6,8,13,79,80,4F,79,D6,88,8
,AF,32,A2,82,3C,A2D
1530 REM
1540 DATA 32,1,FB,CD,16,FD,28,1C,13,FE,2C,28,17,1B,CD,76,2C,DA,23,FD,CD,16,FD,28
,6,FE,2C,C2,23,FD,13,3E,DBD
1550 REM
1560 DATA 1,CD,89,24,3A,A2,82,CD,13,7B,21,A3,82,A7,20,A,8,CD,8E,FE,CA,23,FD,8,36
,FF,D5,11,E6,FA,1,10,EAA
1570 REM
1580 DATA 0,ED,B0,D1,8,FE,8,38,4C,8,21,0,0,1,FF,FF,CD,16,FD,28,2E,FE,2C,28,B,CD,
39,4E,CD,BA,FE,44,DD8
1590 REM
1600 DATA 4D,21,0,0,8,CD,8E,FE,20,1F,8,CD,10,23,FE,2C,20,51,13,CD,39,4E,CD,BA,FE
,A7,ED,42,38,45,28,43,C5B
1610 REM
1620 DATA 23,18,7,8,CD,8E,FE,28,3A,8,ED,43,F8,FA,22,F6,FA,CD,1F,FD,8,21,D,FD,E5,
ED,53,5C,FB,4F,CB,3F,1032
1630 REM
1640 DATA CB,3F,CD,9B,33,2,FE,28,FD,4D,FD,59,FD,ED,5B,5C,FB,1A,13,C3,4F,6D,CD,10
,23,FE,D,C8,FE,3A,C9,CD,1156
1650 REM
1660 DATA 16,FD,C8,3E,23,C3,A1,FB,CD,1F,FD,3E,3,32,E5,FA,21,0,18,22,F6,FA,22,FA,
FA,EB,21,0,20,22,FC,FA,107B
1670 REM
1680 DATA 21,0,0,22,F8,FA,79,FE,5,28,6C,18,16,3E,2,32,E5,FA,79,FE,9,28,5D,18,A,3
E,1,32,E5,FA,79,FE,CB2
1690 REM
1700 DATA D,28,51,CD,CD,FE,ED,5B,11,FB,2A,F8,FA,7C,FE,FF,20,6,BD,20,3,2A,13,FB,C
D,A8,FE,E5,FD,E1,CD,9C,12E4
1710 REM
1720 DATA FE,28,1,37,3E,FF,F5,C5,CD,0,F9,30,1B,C1,F1,8,3A,E5,FA,FE,3,C0,2A,FC,FA
,ED,5B,FA,FA,CD,AE,FE,13CF
1730 REM
1740 DATA E5,FD,E1,8,CD,0,F9,D8,3A,1,FB,A7,CA,0,0,3E,3F,C3,A1,FB,2A,F8,FA,E5,21,
3C,FB,CD,52,4E,CD,6D,11EC
1750 REM
1760 DATA 43,A7,28,FA,AF,32,E4,FA,FD,21,E5,FA,11,1B,0,AF,CD,39,FA,6,32,76,10,FD,
FD,E1,3A,E5,FA,FE,2,3E,1193
1770 REM
1780 DATA FF,38,3,32,E4,FA,ED,5B,F6,FA,F5,CD,39,FA,F1,8,3A,E5,FA,FE,3,C0,FD,2A,F
C,FA,ED,5B,FA,FA,8,C3,156E
1790 REM
1800 DATA 39,FA,AF,32,E5,FA,CD,16,FD,21,0,0,28,18,CD,10,23,FE,86,C2,23,FD,13,79,
```

```
3D,C2,23,FD,CD,39,4E,CD,F66
1810 REM
1820 DATA 1F,FD,ED,53,5C,FB,22,FE,FA,79,3D,28,41,CD,CD,FE,2A,60,81,ED,5B,11,FB,7
9,A7,C2,7B,FD,E5,19,22,66,11C3
1830 REM
1840 DATA 81,D5,ED,5B,68,81,1B,ED,52,3E,1C,D2,A1,FB,D1,C1,2A,13,FB,9,22,62,81,2A
,15,FB,9,22,64,81,60,69,E94
1850 REM
1860 DATA CD,7B,FD,2A,19,FB,7C,B5,CA,B5,6A,C3,8C,67,2A,66,81,ED,5B,60,81,D5,ED,5
2,22,F6,FA,2A,62,81,ED,52,11FF
1870 REM
1880 DATA 22,F8,FA,2A,64,81,ED,52,22,FA,FA,C3,B8,FD,FE,D,C8,FE,B,D0,C5,4F,E6,3,3
D,79,C1,C9,79,FE,E,C8,1326
1890 REM
1900 DATA FE,B,D0,E6,3,FE,2,C9,3A,E5,FA,FE,2,D8,CD,C6,FE,CD,9C,FE,C8,7C,F6,40,67
,C9,8,F5,E6,C,FE,8,1413
1910 REM
1920 DATA CC,C6,FE,F1,8,C9,7C,E6,C0,C2,23,FD,C9,AF,32,E4,FA,C5,FD,21,0,FB,FD,E5,
11,1B,0,AF,37,CD,0,F9,1376
1930 REM
1940 DATA FD,E1,30,F2,3A,0,FB,FE,4,30,EB,1,2,1,1E,F0,B8,28,3,B9,20,A,3A,E5,FA,B8
,28,B,B9,28,8,78,D8F
1950 REM
1960 DATA FD,BE,E5,28,2,1E,80,FD,46,0,4,21,1B,FB,4B,AF,ED,B1,10,FA,CD,52,4E,3E,3
A,CD,52,2B,4B,FD,E5,D1,FB5
1970 REM
1980 DATA 21,E6,FF,19,6,10,7E,3C,20,3,79,80,4F,13,1A,BE,23,20,1,C,CD,52,2B,10,F4
,3E,A,CD,52,2B,3E,D,9C0
1990 REM
2000 DATA CD,52,2B,CB,79,20,8F,C1,3A,E5,FA,FE,2,D8,3E,FF,32,E4,FA,C9,0,0,0,0,0,0
,0,0,0,0,0,0,C05
2010 REM
2020 DATA 21,7,98,11,0,F9,6,8,C5,1,E0,0,ED,B0,1,21,0,9,C1,10,F3,CD,67,FB,C3,26,3
D,0,0,0,0,0,A5F
```

POKER

Another great poker game
from TERRY COLE

```
10 REM  POKER
20 REM  By Terry Cole 1986
30 REM  ------------------
40 REM
50 SCREEN 2,2:COLOR1,15,,15:CLS
60 PRINT CHR$(16)
70 COLOR1,10,(0,0)-(255,191),10
80 COLOR1,2,(130,8)-(255,60),10
90 COLOR1,6,(130,168)-(255,191),10
100 LINE (129,168)-(255,191),1,B
110 LINE (143,80)-(247,152),1,B
120 LINE (129,8)-(255,60),1,B
130 COLOR1,15,(148,80)-(240,152),10
140 LINE (8,8)-(120,176),1,B
150 COLOR1,15,(8,8)-(115,176),10
160 CURSOR 155,170:PRINT " Presented by"
170 CURSOR 134,180:PRINT "TERRY COLE  ROTORUA"
180 LINE (163,100)-(171,140),5,B
190 LINE (153,90)-(181,100),5,B
200 LINE (188,90)-(196,130),5,B
```

```
210 LINE (188,130)-(206,140),5,B
220 LINE (213,90)-(221,140),5,B
230 LINE (221,90)-(231,100),5,B
240 LINE (221,130)-(231,140),5,B
250 PAINT (165,101),5
260 PAINT(155,91),5
270 PAINT(190,91),5
280 PAINT(190,131),5
290 PAINT (223,92),5
300 PAINT(217,91),5
310 PAINT(223,131),5
320 CIRCLE(80,150),35,5,.4,.79,.69
330 LINE (80,10)-(15,150),10
340 LINE (80,10)-(19,150),10
350 LINE (80,10)-(17,150),10
360 LINE (80,10)-(16,150),10
370 LINE (80,10)-(18,150),10
380 LINE (15,150)-(19,150),10
390 CIRCLE (30,130),4,5,,,,BF
400 CIRCLE(14,40),70, 9,1.3,.95,.19
410 LINE (65,145)-(93,145),1
420 LINE (65,145)-(70,135),1
430 LINE (70,135)-(75,131),1
440 LINE (75,131)-(79,129),1
450 LINE (79,129)-(85,135),1
460 LINE (85,135)-(93,145),1
470 PAINT(80,139),1
480 CIRCLE(80,150),29,5,.3,.85,.65
490 CIRCLE(80,149),22,5,.2,.80,.70
500 CIRCLE(26,25),10,11,,,,BF
510 LINE (8,8)-(120,176),1,B
520 PSET (80,150),6
530 PSET (81,149),6
540 PSET (81,150),6
550 PSET (82,150),6
560 CURSOR 130,10:COLOR1:PRINT CHR$(17);" P O K E R "
570 CURSOR 142,25:PRINT " J ÷ ≈ ° "
580 PRINT CHR$(16)
590 CURSOR 140,35:PRINT "A PAIR OF ACES OR"
600 CURSOR 155,45:PRINT "BETTER TO WIN"
610 RESTORE 650
620 FOR J=1 TO 54:READ F:SOUND1,F,14:SOUND2,F+2,14:SOUND3,F*2,14:FOR DE=1 TO 30:
NEXT DE,J
630 READ D
640 IF D=0 THEN SOUND 0
650 DATA 330,294,262,330,294,262,392,349,330,392,349,330,262,330,262,330,262,392
,262,392
660 DATA 392,262,392,262,330,262,330,262,330,523,494,440,523,494,440
670 DATA 659,587,659.523.587.494.523.440,494,392,440,349,392,330,349,294,330,262
,262,0
680 CURSOR  8,180:PRINT "Press a key to begin"
690 IF INKEY$="" THEN 690
700 REM POKER
710 CA=100
720 GOSUB 3110
730 Q1=0:Q2=0:Q3=0:Q4=0:Q5=0
740 P=0
750 IF Q1=1 THEN P=1:GOTO 770
760 GOSUB 1430
770 SCREEN 2,2:CLS
780 U=0
790 COLOR1,8,(0,0)-(255,191),4
800 CURSOR 160,160:PRINT "BET $";BE
810 CURSOR 90,180:PRINT "** POKER **    by TERRY COLE"
820 X=10
830 CURSOR 15,20:PRINT "1"
840 CURSOR 55,20:PRINT "2"
850 CURSOR 95,20:PRINT "3"
860 CURSOR 135,20:PRINT "4"
870 CURSOR 175,20:PRINT "5"
880 LINE (0,28)-(195,28),1
890 LINE (0,50)-(195,50),1
900 CURSOR X,40:PRINT Z$
```

```
910 IF Q2=1 THEN P=1:GOTO 930
920 GOSUB 1580
930 X=X+40
940 CURSOR X,40:PRINT X$
950 GOSUB 2250
960 IF Q3=1 THEN P=1:GOTO 980
970 GOSUB 1730
980 X=X+40
990 CURSOR X,40:PRINT Y$
1000 IF Q4=1 THEN P=1:GOTO 1020
1010 GOSUB 1890
1020 X=X+40
1030 CURSOR X,40:PRINT W$
1040 IF Q5=1 THEN P=1:GOTO 1060
1050 GOSUB 2060
1060 X=X+40
1070 CURSOR X,40:PRINT V$
1080 IF P=1 THEN GOTO 1360
1090 GOSUB 2220
1100 A$=INKEY$
1110 IF A$="1" THENU=1: GOTO 1180
1120 IF A$="2" THENU=1: GOTO 1210
1130 IF A$="3" THENU=1: GOTO 1240
1140 IF A$="4" THENU=1: GOTO 1270
1150 IF A$="5" THENU=1: GOTO 1300
1160 IF A$="0"AND U=1 THEN GOTO 1330
1170 GOTO 1100
1180 CURSOR 10,100:PRINT "HELD":Q1=1
1190 GOSUB 2250
1200 GOTO 1100
1210 CURSOR 50,100:PRINT "HELD":Q2=1
1220 GOSUB 2250
1230 GOTO 1100
1240 CURSOR 90,100:PRINT "HELD":Q3=1
1250 GOSUB 2250
1260 GOTO 1100
1270 CURSOR 130,100:PRINT "HELD":Q4=1
1280 GOSUB 2250
1290 GOTO 1100
1300 CURSOR 170,100:PRINT "HELD":Q5=1
1310 GOSUB 2250
1320 GOTO 1100
1330 CURSOR 60,130:PRINT "RE-DEAL"
1340 GOSUB 2280
1350 FOR I=1 TO 200:NEXT
1360 IF P=1 THEN GOTO 1380
1370 GOTO 750
1380 GOTO 2370
1390 CURSOR 20,130:PRINT "YOU WON $";ZA
1400 GOSUB 2250
1410 CURSOR 20,160:PRINT "YOU HAVE $";CA
1420 GOTO 2310
1430 A=INT(RND(1)*13)+1
1440 IF A=1 THEN Z$="NINE":T=8
1450 IF A=2 THEN Z$="TEN" :T=9
1460 IF A=11 THEN Z$="TWO":T=1
1470 IF A=12 THEN Z$="THREE":T=2
1480 IF A=13 THEN Z$="FOUR":T=3
1490 IF A=3 THEN Z$="JACK":T=10
1500 IF A=4 THEN Z$="QUEEN":T=11
1510 IF A=5 THEN Z$="KING":T=12
1520 IF A=9 THEN Z$="SIX":T=5
1530 IF A=10 THEN Z$="FIVE":T=4
1540 IF A=6 THEN Z$="ACE":T=13
1550 IF A=7 THEN Z$="EIGHT":T=7
1560 IF A=8 THEN Z$="SEVEN":T=6
1570 RETURN
1580 A=INT(RND(1)*13)+1
1590 IF A=1 THEN X$="NINE":SA=8
1600 IF A=2 THEN X$="TEN":SA=9
1610 IF A=11 THEN X$="TWO":SA=1
1620 IF A=12 THEN X$="THREE":SA=2
1630 IF A=13 THEN X$="FOUR":SA=3
```

```
1640 IF A=3 THEN X$="JACK":SA=10
1650 IF A=4 THEN X$="QUEEN":SA=11
1660 IF A=5 THEN X$="KING":SA=12
1670 IF A=9 THEN X$="SIX":SA=5
1680 IF A=10 THEN X$="FIVE":SA=4
1690 IF A=6 THEN X$="ACE":SA=13
1700 IF A=7 THEN X$="EIGHT":SA=7
1710 IF A=8 THEN X$="SEVEN":SA=6
1720 RETURN
1730 A=INT(RND(1)*13)+1
1740 IF A=1 THEN Y$="NINE":SB=8
1750 PRINT :PRINT
1760 IF A=2 THEN Y$="TEN":SB=9
1770 IF A=11 THEN Y$="TWO":SB=1
1780 IF A=12 THEN Y$="THREE":SB=2
1790 IF A=13 THEN Y$="FOUR":SB=3
1800 IF A=3 THEN Y$="JACK":SB=10
1810 IF A=4 THEN Y$="QUEEN":SB=11
1820 IF A=5 THEN Y$="KING":SB=12
1830 IF A=9 THEN Y$="SIX":SB=5
1840 IF A=10 THEN Y$="FIVE":SB=4
1850 IF A=6 THEN Y$="ACE":SB=13
1860 IF A=7 THEN Y$="EIGHT":SB=7
1870 IF A=8 THEN Y$="SEVEN":SB=6
1880 RETURN
1890 A=INT(RND(1)*13)+1
1900 IF A=1 THEN W$="NINE":SC=8
1910 IF A=2 THEN W$="TEN":SC=9
1920 IF A=11 THEN W$="TWO":SC=1
1930 IF A=12 THEN W$="THREE":SC=2
1940 IF A=13 THEN W$="FOUR":SC=3
1950 IF A=3 THEN W$="JACK":SC=10
1960 IF A=4 THEN W$="QUEEN":SC=11
1970 IF A=5 THEN W$="KING":SC=12
1980 IF A=9 THEN W$="SIX":SC=5
1990 IF A=10 THEN W$="FIVE":SC=4
2000 IF A=6 THEN W$="ACE":SC=13
2010 IF A=7 THEN W$="EIGHT":SC=7
2020 IF A=8 THEN W$="SEVEN":SC=6
2030 RETURN
2040 IF A=7 THEN W$="EIGHT"
2050 IF A=8 THEN W$="SEVEN"
2060 A=INT(RND(1)*13)+1
2070 IF A=1 THEN V$="NINE":SE=8
2080 IF A=2 THEN V$="TEN":SE=9
2090 IF A=11 THEN V$="TWO":SE=1
2100 IF A=12 THEN V$="THREE":SE=2
2110 IF A=13 THEN V$="FOUR":SE=3
2120 IF A=3 THEN V$="JACK":SE=10
2130 IF A=4 THEN V$="QUEEN":SE=11
2140 IF A=5 THEN V$="KING":SE=12
2150 IF A=9 THEN V$="SIX":SE=5
2160 IF A=10 THEN V$="FIVE":SE=4
2170 IF A=6 THEN V$="ACE":SE=13
2180 IF A=7 THEN V$="EIGHT":SE=7
2190 IF A=8 THEN V$="SEVEN":SE=6
2200 RETURN
2210 GOTO 2210
2220 CURSOR 20,140:PRINT "PRESS 1/2/3/4/5 TO HOLD CARDS"
2230 CURSOR 20,160 :PRINT "PRESS 0 TO RE-DEAL"
2240 RETURN
2250 FOR K=1000 TO 2000 STEP 100
2260 SOUND 1,K,15:NEXT K
2270 SOUND 0:RETURN
2280 FOR K=2000 TO 1000 STEP -100
2290 SOUND 1,K,15:NEXT K
2300 SOUND 0:RETURN
2310 IF CA<1 THEN GOTO 3250
2320 A$=INKEY$
2330 CURSOR 20,110:PRINT "ANOTHER TURN Y/N"
2340 IF A$="Y" THEN GOSUB 2280:GOTO 720
2350 IF A$="N" THEN END
2360 GOTO 2310
2370 IF Z$=X$ AND Y$=X$ AND W$=Y$ AND V$=W$  THEN GOTO 2910
```

```
2380 IF Z$=X$ AND X$=Y$ AND Y$=W$ THEN GOTO 2920
2390 IF Z$=X$ AND X$=Y$ AND Y$=V$ THEN GOTO 2920
2400 IF Z$=X$ AND X$=W$ AND W$=V$ THEN GOTO 2920
2410 IF Z$=Y$ AND Y$=W$ AND W$=V$ THEN GOTO 2920
2420 IF X$=Y$ AND Y$=W$ AND W$=V$ THEN GOTO 2920
2430 IF Z$=X$ AND X$=Y$ AND W$=V$ THEN GOTO 2930
2440 IF Z$=Y$ AND Y$=W$ AND X$=V$ THEN GOTO 2930
2450 IF X$=Y$ AND Y$=W$ AND Z$=V$ THEN GOTO 2930
2460 IF Y$=W$ AND W$=V$ AND Z$=X$ THEN GOTO 2930
2470 IF Z$=W$ AND W$=V$ AND X$=Y$ THEN GOTO 2930
2480 IF Z$=X$ AND X$=W$ AND Y$=V$ THEN GOTO 2930
2490 IF X$=Y$ AND Y$=V$ AND Z$=W$ THEN GOTO 2930
2500 IF Z$=X$ AND X$=V$ AND Y$=W$ THEN GOTO 2930
2510 IF Z$=Y$ AND Y$=V$ AND X$=W$ THEN GOTO 2930
2520 IF X$=W$ AND W$=V$ AND Z$=Y$ THEN GOTO 2930
2530 GOSUB 3320
2540 IF Z$=X$ AND X$=Y$ THEN GOTO 2950
2550 IF Z$=Y$ AND Y$=W$ THEN GOTO 2950
2560 IF X$=Y$ AND Y$=W$ THEN GOTO 2950
2570 IF Y$=W$ AND W$=V$ THEN GOTO 2950
2580 IF Z$=W$ AND W$=V$ THEN GOTO 2950
2590 IF Z$=X$ AND X$=W$ THEN GOTO 2950
2600 IF Z$=Y$ AND Y$=V$ THEN GOTO 2950
2610 IF X$=Y$ AND Y$=V$ THEN GOTO 2950
2620 IF X$=W$ AND W$=V$ THEN GOTO 2950
2630 IF Z$=X$ AND X$=V$ THEN GOTO 2950
2640 IF Z$=X$ AND Y$=W$ THEN GOTO 2960
2650 IF Z$=X$ AND Y$=V$ THEN GOTO 2960
2660 IF Z$=Y$ AND W$=V$ THEN GOTO 2960
2670 IF Z$=Y$ AND X$=W$ THEN GOTO 2960
2680 IF Z$=W$ AND X$=Y$ THEN GOTO 2960
2690 IF Z$=W$ AND X$=V$ THEN GOTO 2960
2700 IF Z$=W$ AND V$=Y$ THEN GOTO 2960
2710 IF Z$=X$ AND V$=W$ THEN GOTO 2960
2720 IF Z$=Y$ AND V$=X$ THEN GOTO 2960
2730 IF Z$=V$ AND X$=Y$ THEN GOTO 2960
2740 IF Z$=V$ AND X$=W$ THEN GOTO 2960
2750 IF Z$=V$ AND W$=Y$ THEN GOTO 2960
2760 IF X$=W$ AND V$=Y$ THEN GOTO 2960
2770 IF X$=V$ AND W$=Y$ THEN GOTO 2960
2780 IF X$=Y$ AND W$=V$ THEN GOTO 2960
2790 IF Y$=W$ AND X$=V$ THEN GOTO 2960
2800 IF T=13 AND SA=13 THEN GOTO 2970
2810 IF T=13 AND SC=13 THEN GOTO 2970
2820 IF T=13 AND SB=13 THEN GOTO 2970
2830 IF T=13 AND SE=13 THEN GOTO 2970
2840 IF SA=13 AND SB=13 THEN GOTO 2970
2850 IF SA=13 AND SC=13  THEN GOTO 2970
2860 IF SA=13 AND SE=13  THEN GOTO 2970
2870 IF SB=13 AND SC=13 THEN GOTO 2970
2880 IF SB=13 AND SE=13 THEN GOTO 2970
2890 IF SC=13 AND SE=13 THEN GOTO 2970
2900 GOTO 2980
2910 CURSOR 50 ,70:PRINT "FIVE OF A KIND":GOTO 3030
2920 CURSOR 50 ,70:PRINT "FOUR OF A KIND":GOTO 3040
2930 CURSOR 50,70:PRINT "FULL HOUSE":GOTO 3050
2940 CURSOR 50,70:PRINT "STRAIGHT":GOTO 3080
2950 CURSOR 50,70:PRINT "THREE OF A KIND ":GOTO 3060
2960 CURSOR 50,70:PRINT "TWO PAIR":GOTO 3070
2970 CURSOR 50,70:PRINT "PAIR OF ACES":GOTO 3090
2980 CURSOR 50,60:PRINT "YOU LOST"
2990 FOR K=1000 TO 200 STEP -100
3000 SOUND 1,K,15:NEXT K
3010 SOUND 0
3020 GOTO 3100
3030 ZA=BE+(BE*5):CA=CA+ZA:GOTO 1390
3040 ZA=BE+(BE*4):CA=CA+ZA:GOTO 1390
3050 ZA=BE+(BE*3):CA=CA+ZA:GOTO 1390
3060 ZA=BE+(BE*2):CA=CA+ZA:GOTO 1390
3070 ZA=BE+(BE*1.5):CA=CA+ZA:GOTO 1390
3080 ZA=BE+(BE*2.5):CA=CA+ZA:GOTO 1390
3090 ZA=BE+(BE*.50):CA=CA+ZA:GOTO 1390
3100 ZA=0:GOTO 1390
```

```
3110 SCREEN 1,1:CLS
3120 COLOR1,8
3130 PRINT:PRINT "          J÷≈° POKER °≈÷J"
3140 PRINT :PRINT
3150 IF CA=100 THEN PRINT "HI THERE GAMBLER!"
3160 IF CA<100 AND  CA >50 THEN PRINT "YOU AIN'T DOING TOO GOOD!!"
3170 IF CA<50 THEN PRINT "NEXT BET SUCKER!!"
3180 IF CA>100 AND CA<500 THEN PRINT "YOU ARE DOING WELL!"
3190 IF CA>500 THEN PRINT "WHEN ARE YOU GOING HOME?"
3200 PRINT :PRINT "YOU HAVE $";CA
3210 PRINT :INPUT "HOW MUCH DO YOU WANT TO BET   ";BE
3220 IF BE>CA THEN 3210
3230 CA=CA-BE
3240 RETURN
3250 SCREEN 2,2:CLS
3260 CURSOR 58,60:PRINT "YOU ARE BROKE SUCKER"
3270 PRINT :PRINT
3280 PRINT "          ANOTHER TURN (Y/N)"
3290 IF INKEY$="Y" THEN 700
3300 IF INKEY$="N" THEN END
3310 GOTO 3280
3320 SS=SA+SB+SC+SE
3330 IF Z$=X$ THEN RETURN
3340 IF X$=Y$ THEN RETURN
3350 IF Y$=W$ THEN RETURN
3360 IF W$=V$ THEN RETURN
3370 IF Z$=Y$ THEN RETURN
3380 IF Z$=W$ THEN RETURN
3390 IF Z$=V$ THEN RETURN
3400 IF X$=W$ THEN RETURN
3410 IF X$=V$ THEN RETURN
3420 IF Y$=V$ THEN RETURN
3430 IF T=1 AND SS=14 THEN GOTO 2940
3440 IF T=2 AND SS=18 THEN GOTO 2940
3450 IF T=2 AND SS=13 THEN GOTO 2940
3460 IF T=3 AND SS=22 THEN GOTO 2940
3470 IF T=3 AND SS=17 THEN GOTO 2940
3480 IF T=3 AND SS=12 THEN GOTO 2940
3490 IF T=4 AND SS=26 THEN GOTO 2940
3500 IF T=4 AND SS=21 THEN GOTO 2940
3510 IF T=4 AND SS=16 THEN GOTO 2940
3520 IF T=4 AND SS=11 THEN GOTO 2940
3530 IF T=5 AND SS=30 THEN GOTO 2940
3540 IF T=5 AND SS=25 THEN GOTO 2940
3550 IF T=5 AND SS=20 THEN GOTO 2940
3560 IF T=5 AND SS=15 THEN GOTO 2940
3570 IF T=5 AND SS=10 THEN GOTO 2940
3580 IF T=6 AND SS=34 THEN GOTO 2940
3590 IF T=6 AND SS=29 THEN GOTO 2940
3600 IF T=6 AND SS=24 THEN GOTO 2940
3610 IF T=6 AND SS=19 THEN GOTO 2940
3620 IF T=6 AND SS=14 THEN GOTO 2940
3630 IF T=7 AND SS=38 THEN GOTO 2940
3640 IF T=7 AND SS=33 THEN GOTO 2940
3650 IF T=7 AND SS=28 THEN GOTO 2940
3660 IF T=7 AND SS=23 THEN GOTO 2940
3670 IF T=7 AND SB=1 THEN RETURN
3680 IF T=7 AND SS=18 THEN GOTO 2940
3690 IF T=8 AND SS=42 THEN GOTO 2940
3700 IF T=8 AND SS=37 THEN GOTO 2940
3710 IF T=8 AND SS=32 THEN GOTO 2940
3720 IF T=8 AND SS=27 THEN GOTO 2940
3730 IF T=8 AND SS=22 THEN GOTO 2940
3740 IF T=9 AND SS=46 THEN GOTO 2940
3750 IF T=9 AND SS=41 THEN GOTO 2940
3760 IF T=9 AND SS=36 THEN GOTO 2940
3770 IF T=9 AND SS=31 THEN GOTO 2940
3780 IF T=9 AND SS=26 THEN GOTO 2940
3790 IF T=10 AND SS=45 THEN GOTO 2940
3800 IF T=10 AND SS=40 THEN GOTO 2940
3810 IF T=10 AND SS=35 THEN GOTO 2940
3820 IF T=10 AND SS=30 THEN GOTO 2940
3830 IF T=11 AND SS=44 THEN GOTO 2940
```

```
3840 IF T=11 AND SS=39 THEN GOTO 2940
3850 IF T=11 AND SS=34 THEN GOTO 2940
3860 IF T=12 AND SS=43 THEN GOTO 2940
3870 IF T=12 AND SS=38 THEN GOTO 2940
3880 IF T=13 AND SS=42 THEN GOTO 2940
3890 RETURN
```

DISK FILE REMOVER

By MICHAEL HADRUP (C) 1986

```
10 REM Disk File Remover
20 REM
30 REM By MICHAEL HADRUP (C) 1986
40 REM
50 REM
60 REM Used to remove unwanted files
70 REM from Disk
80 REM Enter "Y" to remove file
90 REM
100 GOTO220
110 FORM=1TO128STEP16
120 IFASC(MID$(Z$,M))=0THEN200
130 F=F+1:K=0
140 C=ASC(MID$(F$,ASC(MID$(Z$,M+12))+1))
150 IFC<160THENK=K+4:C=ASC(MID$(F$,C+1)):GOTO150
160 K=K+1*(C-192)
170 PRINTMID$(Z$,M,12);" ";K;TAB(18);"Sector(s)";TAB(28);"REMOVE Y/N";
180 A$=INKEY$:IFA$=""THEN180
190 IFA$="Y"ORA$="y"THENKILLMID$(Z$,M,12):D=D+1
200 NEXTM
210 RETURN
220 COLOR15,1:CLS:F=0:D=0
230 PRINTTAB(6);"::: Disk File Remover :::"
240 PRINT:PRINT"Insert DISK and press SPACE":PRINT
250 A$=INKEY$:IFA$<>" "THEN250
260 DSKI$20,13;F$,0,160
270 FORN=1TO12
280 DSKI$20,N;A$,0,128;B$,128,128
290 Z$=A$:GOSUB110
300 Z$=B$:GOSUB110
310 NEXTN
320 PRINT:PRINT"Files deleted:";D,"Files remaining";F-D
330 PRINT:PRINT"Disk free:";DSKF;"K"
340 PRINT:PRINT"Another DISK (Y/N) ?"
350 A$=INKEY$:IFA$=""THEN350
360 IFA$="Y"ORA$="y"THEN220
370 CALL &H623C:REM NEW
```

USER GROUPS

AUCKLAND (NTH SHORE) SEGA USER GROUP (1)
Contact: Andrew Flexman
426 Glenfield Rd
Glenfield
AUCKLAND
Ph: 444-8244

AUCKLAND (NTH SHORE) SEGA USER GROUP (2)
Contact: Norman Raynel
78 Anzac St
Takapuna
AUCKLAND
Ph: 496-841

PUKEKOHE SEGA USER GROUP
Contact: Selwyn Easton
4 Roose Ave
PUKEKOHE
Ph: 86-583

SOUTH COROMANDEL SEGA USER GROUP
Contact: Sid Hawken
PO Box 183
WHANGAMATA
Ph: 58-775

HAMILTON SEGA USER GROUP
PO Box 1548 Hamilton
President: Mr Colin Bell Ph: 73-826
Secretary: Mrs Anne Thrush Ph: 437-312
Meetings held 2nd & 4th Monday of each month at Whitiora Community Centre, Willoughby St at 7.30pm.

TOKOROA SEGA USER GROUP
Contact: Geoff Crawford
1 Pio Pio Pl
TOKOROA
Ph: 67-105

ROTORUA SEGA USER GROUP
Contact: Terry Cole
PO Box 7140
Te Ngae
ROTORUA
Ph: 59-325 AH

WHAKATANE SEGA USER GROUP
Contact: K Nightingale
240 King St
WHAKATANE
Ph: 84-500

NAPIER SEGA USER GROUP
Contact: Reid Duncan
Store Manager
Agnews Refrigeration
Ph: 55-857, 57-431

HAWERA SEGA USER GROUP
Contact: D.M. Beale
7A Clive St
HAWERA
Ph: 85-108

MARTON SEGA USER GROUP
Contact: Mr H.R. Miller
41 Alexandra St
MARTON

WELLINGTON SEGA USER GROUP
Contact: Shaun Parsons
PO Box 1871
WELLINGTON
Ph: 897-095 (AH) 727-666 (BUS)

CHRISTCHURCH SEGA USER GROUP
Contact: James O'Donnell
15 Jebson St
Shirley
CHRISTCHURCH
Ph: 856-884

TIMARU SEGA USER GROUP
Contact: John Oliver
South Canterbury Computer User Group
PO Box 73
TIMARU
Ph 26-300

OAMARU SEGA USER GROUP
Contact: Bill Dowman
99 Aln St
OAMARU
Ph: 46-250

DUNEDIN SEGA USER GROUP
Central City Computer Interest Group
Box 5260
Moray Place
DUNEDIN
Contact: Graeme Simpson,
Saddle Hill, R.D.1, Dunedin.
Ph: (089) 6374

HI-SCORE CHALLENGE

Borderline	10,389,999	Michael Wilkinson
Congo Bongo	139,320	Paul Noyes
Flicky	230,300	Brendan Hallett
Lode Runner	38,000	Steve Biggs
Monaco GP	210,062	Stewart Parkes
Pacar	1,495,500	Tony Sasso
Pop Flamer	117,000	Richard Hendra
Sinbad Mystery	190,000	Stewart Parkes
Star Jacker	398,000	Steve Biggs
Video Flipper	999,880	Andre Stokes
Yamoto	85,000	Stewart Parkes
N-Sub	76,250	Reuban Jackson
Safari Hunting	4,300	Richard Hendra
Safari Race	30,000	Jonathan Fletcher

Challenge these Hi-Scores for Cartridge games by sending us yours.

OVERSEAS SEGA USERS CLUBS

Klaus Pinker
P.O. Box 18
Belconnen
ACT 2616
Ph: 062 30 2334

Scott McDonald
2 Coolalie Ave
Camden
NSW 2570
Ph: 046 668 956

John McLennan
65 Highclere Blvd
Marangaroo
WA 6064
Ph: 09 342 3905

Jan Jacobson
10 Pioneer Ave
O'Sullivans Beach
SA 5166
Ph: 08 382 7967

Les Beacall
1/41 Cameron Road
Croydon
VIC 3136
Ph: 03 725 0864

Wayne Ariel
35 Leanne Street
Marsden
QLD 4203